



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Modified Position Update in Spider Monkey Optimization Algorithm

¹Sandeep Kumar, ²Vivek Kumar Sharma, ³Rajani Kumari

Faculty of Engineering and Technology
Jagannath University, Chaksu, Jaipur-303901
INDIA

Abstract: Spider Monkey optimization (SMO) algorithm is newest addition in class of swarm intelligence. SMO is a population based stochastic meta-heuristic. It is motivated by intelligent foraging behaviour of fission fusion structured social creatures. SMO is a very good option for complex optimization problems. This paper proposed a modified strategy in order to enhance performance of original SMO. This paper introduces a position update strategy in SMO and modifies both local leader and global leader phase. The proposed strategy is named as Modified Position Update in Spider Monkey Optimization (MPU-SMO) algorithm. The proposed algorithm tested over benchmark problems and results show that it gives better results for considered unbiased problems.

Keywords: Spider Monkey Optimization, Swarm Intelligence, Population based Metaheuristics, Modified Position Update in Spider Monkey Optimization

I. Introduction

Nature inspired meta-heuristics has turn out to be an intimidating and out of the ordinary field of research in the midst of researchers who are trying to solve complex optimization problems. More or less all meta-heuristics make use of both randomization and local search. Due to randomization it can be in motion away from local search to global search space. That's why meta-heuristics are unsurpassed more suitable for global optimization problems. Meta heuristic algorithms have two major components: diversification and intensification. Diversification is the process of exploration of the large search space and ensures that solution does not ensnare in local optima at the same time as intensification concentrates on best solution for convergence to optimality [1]. Population based meta-heuristics do not give assurance for the optimal solution but they provide near-optimal solution for most difficult optimization problems. Researchers have evaluated this type of behaviors and developed strategies with the intention of can be used to solve nonlinear and discrete optimization problems. Preceding research [2, 3, 4, and 5] in last decade has shown that strategies based on swarm intelligence have enormous prospective to come across solutions of real world optimization problems. The algorithms that have emerged in topical years consist of ant colony optimization (ACO) [2], particle swarm optimization (PSO) [3], bacterial foraging optimization (BFO) [6], Artificial bee colony (ABC) optimization algorithm established by D. Karaboga [7] and most recently developed Spider Monkey Optimization (SMO) algorithm [10] is new entry in class of swarm intelligence. This SMO algorithm is inspired by fission fusion social structure (FFSS) based foraging behavior of spider monkeys when searching for quality food source and for mating. Similar to any other population based optimization techniques, ABC consists of a population of inherent solutions. The inherent solutions are food sources of honey bees. The fitness is decided in terms of the quality of the food source that is nectar amount. ABC is relatively a straightforward, speedy and population based stochastic search technique in the field of nature inspired algorithms. SMO is also similar to ABC in nature.

There are two fundamental processes which drive the swarm to update in ABC: the deviation process, which enables exploring different fields of the search space, and the selection process, which ensures the exploitation of the previous experience. However, it has been shown that the ABC may occasionally stop moving toward the global optimum even though the population has not encounter to a local optimum [8]. It can be observed that the solution search equation of ABC algorithm is good at exploration but poor at exploitation [9]. Therefore, to maintain the proper balance between exploration and exploitation behavior of ABC, it is highly expected to develop a local search approach in the basic ABC to intensify the search region.

II. Spider Monkey Optimization (SMO) Algorithm

Social activities of spider monkeys encouraged JC Bansal et al. [10] to develop a stochastic optimization modulus operandi that impersonate fission-fusion social structure (FFSS) based intelligent foraging behavior of spider monkeys. JC Bansal et al. [10] identified following four key features of the FFSS.

- ❖ The fission-fusion social organization based animals are societal and survive in groups of 40-50 individuals. The FFSS of swarm may diminish the foraging antagonism among group members by separating them into sub-groups in order to search food [10].
- ❖ A most senior female in general leads the group and is conscientious for searching food sources. It is denoted as global leader. If she is not competent to search out an adequate amount of food for the group,

she divides the group into minor subgroups (size may vary from 3 to 8 individuals) that forage autonomously.

- ❖ Sub-groups are also believed to be leaded by a female who becomes decision-maker for planning a well-organized foraging route each day. Here this leader is known as local leader [10].
- ❖ The group members exchange a few words among themselves and with other group members, to uphold social bonds and protective boundaries [10].

In the SMO algorithm, foraging behavior of FFSS based flora and fauna (like spider monkeys) is alienated into four stages.

Step 1. The group starts food foraging and appraises their distance from the food.

Step 2. Group members bring up to date their positions based on the distance from the foods source and all over again appraise distance from the food sources.

Step 3. Moreover, in this step, the local leader updates its best location within the group and if the location is not updated for a predefined number of times then all members of that group start searching of the food sources in different directions.

Step 4. Subsequently, in the last step, the global leader keep informed its eternally best position and in case of stagnation, it divides the group into smaller size subgroups.

Above described four steps mentioned aforesaid, are constantly executed until the preferred output is achieved. It is observed that in SMO algorithm there are two most important control parameters are Global Leader Limit (GL_{limit}) and Local Leader Limit (LL_{limit}) which provide appropriate direction to global and local leaders respectively. In SMO stagnation can be avoided by using LL_{limit} . If a local group leader does not keep informed her-self after a predefined number of times then that group is re-directed to another direction for in order to search food. Here, the term predefined number of times is referred as LL_{limit} . An additional control parameter, that is to say Global Leader Limit (GL_{limit}) is used for the identical intention by global leader. The global leader divides the group into smaller sub-groups if she does not update in a predefined number of times that is GL_{limit} [10].

A. Analogy between SMO and Swarm intelligence Behaviour

The SMO algorithm also follows self-organization and division of labor properties for obtaining intelligent swarming behaviors of natural world.

Self Organization: Self organization includes positive feedback, negative feedback, fluctuations and multiple interactions [11].

Positive Feedback: As monkeys keep updating their locations by learning from local leader, global leader and self experience in the first and subsequent steps of SMO algorithm, it shows encouraging criticism mechanisms of self-organization.

Negative Feedback: Local leader limit and Global leader limit provides negative feedback to help local and global leader's for their decisions.

Fluctuations: The 3rd step in which the languished group members are redirected to different directions for searching food sources, shows fluctuations characteristic.

Multiple Interactions: As each and every monkey in both global and local leader phase communicates with others it shows multiple interaction property.

Division of Labor: In the 4th step, at what time the global leader is gets trapped, it splits the groups into minor subgroups for the purpose of food foraging. This phenomenon mimics division of labour property of spider monkeys [10].

B. Major steps of Spider Monkey Optimization (SMO) Algorithm

Analogous to the other population-based algorithms, SMO is also a hit and trial based mutual iterative strategy. The SMO progression consists of seven major phases. The detailed description of each step of SMO accomplishment is outlined below:

1) Initialization of the Population

At the outset, SMO engenders an unvaryingly disseminated early population of N spider monkeys where each monkey SM_i ($i = 1, 2, \dots, N$) is a vector of dimension D. At this point D is the number of variables in the optimization problem and SM_i represent the position of i^{th} Spider Monkey (SM) in the population. Each spider monkey SM corresponds to the potential solution of the problem under consideration. Each SM_i is initialized as follows:

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

Here $SM_{\min j}$ and $SM_{\max j}$ are lower and upper bounds of SM_i in j^{th} direction respectively.

2) Local Leader Phase (LLP)

The second phase in SMO is Local Leader phase. In this phase SM update its existing location based on the information from the local leader understanding as well as local group members understanding. The fitness value of so obtained new location is calculated. If the fitness value of the new location is higher than that of the

previous location, subsequently the SM updates his location with the new one. The location modernizes equation for i^{th} SM (which is a member of k^{th} local group) in this phase is as follow:

$$SM_{newij} = SM_{ij} + \phi_1 \times (LL_{kj} - SM_{ij}) + \phi_2 \times (SM_{rj} - SM_{ij}) \quad \text{Here } \phi_1 \in (0,1) \text{ and } \phi_2 \in (-1,1) \quad (2)$$

Where SM_{ij} is the j^{th} dimension of the i^{th} SM, LL_{kj} represents the j^{th} dimension of the k^{th} local group leader position. SM_{rj} is the j^{th} dimension of the r^{th} SM which is chosen randomly within k^{th} group such that $r \neq i$.

3) Global Leader Phase (GLP)

After achievement of the Local Leader phase next phase is Global Leader phase (GLP). During GLP phase, all the SM's bring up to date their location by means of understanding of Global Leader and local group member's understanding. The location modernizes equation for this phase is as follows:

$$SM_{newij} = SM_{ij} + \phi_1 \times (GL_j - SM_{ij}) + \phi_2 \times (SM_{rj} - SM_{ij}) \quad \text{Where } \phi_1 \in (0,1) \quad \phi_2 \in (-1,1) \quad (3)$$

Where GL_j stands for the j^{th} dimension of the global leader location and $j \in \{1, 2, \dots, D\}$ is the haphazardly preferred index.

In GLP phase, the locations of spider monkeys (SM_i) are updated based on probabilities p_i 's which are considered using their fitness. In this way a better candidate will have more chance to make itself better. The probability p_i may be calculated using following expression (there may be some other but must be a function of fitness):

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

Here $fitness_i$ is the fitness value of the i^{th} SM and $fitness_{max}$ is the maximum fitness in the group. Further, the fitness of the newly generated position of the SM's is calculated and compared with the old one and adopted the better position.

4) Global Leader Learning (GLL) phase

In GLL phase, the location of the global leader is modernized by applying the voracious selection approach in the population i.e., the location of the SM having most excellent fitness in the population is selected as the updated location of the global leader. Additional, it is checked that the location of global leader is updating or not and if not then the Global Limit Count is incremented by 1.

5) Local Leader Learning (LLL) phase

In LLL phase, the location of the local leader is updated by applying the greedy selection in that group i.e., the location of the SM having unsurpassed fitness in that group is preferred as the modernized location of the local leader. Next, the updated location of the local leader is compared with the older one and if the local leader is not updated then the Local Limit Count is incremented by 1.

6) Local Leader Decision (LLD) phase

If any Local Leader location is not updated up to a predefined threshold called Local Leader Limit (LL_{limit}), then all the members of that group modernize their locations either by arbitrary initialization or by using mutual information from Global Leader and Local Leader through equation (5), based on the pr (perturbation rate).

$$SM_{newij} = SM_{ij} + \phi \times (GL_j - SM_{ij}) + \phi \times (SM_{ij} - LL_{kj}) \quad \text{Where } \phi \in (0,1) \quad (5)$$

It is understandable from the equation (5) that the updated measurement of this SM is attracted towards global leader and fends off from the local leader.

7) Global Leader Decision (GLD) phase

In GLD phase, the location of global leader is monitored and if it is not updated up to a predefined number of iterations that is known as Global Leader Limit (GL_{limit}), then the global leader divides the population into smaller groups. Firstly, the population is divided into two groups and then three groups and so on till the maximum number of groups (MG) are formed. Each time in GLD phase, LLL process is initiated to decide on the local leader in the recently fashioned groups. The case in which maximum number of groups is formed and even then the position of global leader is not updated then the global leader combines all the groups to form a single group. As a consequence the anticipated algorithm impersonates fusion-fission structure of SMs. The absolute pseudo-code of the SMO algorithm is outlined as follow [10]:

Algorithm 1: Spider Monkey Optimization (SMO) Algorithm:

Step 1. Initialize Population, Local Leader Limit (LL_{limit}), Global Leader Limit (GL_{limit}) and Perturbation rate (pr).
 Step 2. Compute fitness (The distance of each individual from corresponding food sources).
 Step 3. Select leaders (global and local both) by applying greedy selection.
 Step 4. while (Annihilation criteria is not fulfilled) do
 Step 5. Generate the new locations for all the group members by using self experience, local leader experience and group member's experience. Using Equ. (2)
 Step 6. Apply the gluttonous selection process between existing location and newly generated location, based on fitness and select the better one;
 Step 7. Calculate the probability p_i for all the group members using Equ. (4).
 Step 8. Generate new locations for the all the group members, selected by p_i , by using self experience, global leader experience and group members experiences Using Equ. (3)
 Step 9. Update the position of local and global leaders, by applying the greedy selection process on all the groups.
 Step 10. If any Local group leader is not updating her position after a specified number of times (LL_{Limit}) then re-direct all members of that particular group for foraging by algorithm using Equ. (5).
 Step 11. If Global Leader is not updating her position for a specified number of times (GL_{Limit}) then she divides the group into smaller groups by following steps.
 Step 12. End While

III. Modified Position Update in Spider Monkey Optimization Algorithm

Exploration of the complete search space and exploitation of the best solutions found in its proximity may be balanced by maintaining the diversity in local leader and global leader phase of SMO. In order to balance exploration and exploitation of local search space the proposed algorithm modify both local leader phase and global leader phase using modified golden section search GSS [15] method inspired by memetic search in ABC [12], Randomized memetic search in ABC [13] and memetic search in DE [14]. The golden section search first time incorporated in population algorithm by JC Bansal et al [12] in memetic search in ABC (MeABC). MeABC introduced a new search phase in ABC inspired by Golden Section Search (GSS) [15]. In MeABC only the best particle of the current swarm updates itself in its proximity. Original GSS method does not use any gradient information of the function to finds the optima of a uni-modal continuous function. GSS processes the interval $[a = -1.2, b = 1.2]$ and initiates two intermediate points:

$$F_1 = b - (b - a) \times \psi, \quad (6)$$

$$F_2 = a + (b - a) \times \psi, \quad (7)$$

Here $\psi = 0.618$ is the golden ratio.

The detailed GSS process [12] is described in Algorithm 2 as follow:

Algorithm 2: Golden Section Search process

Input: Optimization function $\min f(x)$ s.t. $a \leq x \leq b$ and termination criteria

Repeat while termination criteria fulfill

Calculate F_1 and F_2 as follow

$F_1 = b - (b - a) * \Psi$ and $F_2 = a + (b - a) * \Psi$ here $a = -1.2$, $b = 1.2$ and $\Psi = 0.618$ (Golden ratio)

Calculate $f(F_1)$ and $f(F_2)$

If $f(F_1) < f(F_2)$ then

$b = F_2$ and the solution fall in range $[a, b]$

else

$a = F_1$ and the solution fall in range $[a, b]$

end if

end while

The proposed strategy modify equation (2) and (3) in the following manner. Here f is determined by GSS process as outlined in above algorithm. Position update in local leader phase is done using Equ (8).

$$SM_{newij} = SM_{ij} + \phi_1 \times (LL_{kj} - SM_{ij}) + \phi_2 \times (SM_{rj} - SM_{ij}) + f \times (SM_{rj} - SM_{ij}) \text{ Here } \phi_1 \in (0,1), \phi_2 \in (-1,1), f \text{ decided by GSS} \dots (8)$$

Position update in local leader phase is done using Equ (9).

$$SM_{newij} = SM_{ij} + \phi_1 \times (LL_{kj} - SM_{ij}) + \phi_2 \times (SM_{rj} - SM_{ij}) + f \times (SM_{rj} - SM_{ij}) \text{ Here } \phi_1 \in (0,1), \phi_2 \in (-1,1), f \text{ decided by GSS} \dots (9)$$

The detailed modified position update in SMO algorithm is outlined in algorithm 3. The proposed algorithm tries to balance the exploration and exploitation process by controlling step size.

Algorithm 3 Modified Position Update in Spider Monkey Optimization (SMO) Algorithm:

- Step 1. Initialize Population, Local Leader Limit (LL_{limit}), Global Leader Limit (GL_{limit}) and Perturbation rate (pr).
 Step 2. Compute fitness (The distance of each individual from corresponding food sources).
 Step 3. Select leaders (global and local both) by applying greedy selection.
 Step 4. while (extermination criteria is not fulfilled) do
 Step 5. For finding the objective (Food Source), generate the new locations for all the group members by using self experience, local leader experience and group member's experience.

$$SM_{newij} = SM_{ij} + \phi_1 \times (LL_{kj} - SM_{ij}) + \phi_2 \times (SM_{rj} - SM_{ij}) + \phi_3 (SM_{rj} - SM_{ij}) \text{ Here } \phi_1 \in (0,1), \phi_2 \in (-1,1), \phi_3 \text{ decided by GSS}$$

- Step 6. Apply the gluttonous selection process between existing location and newly generated location, based on fitness and select the better one;

- Step 7. Calculate the probability p_i for all the group members using. $p_i = 0.9 \times \frac{fitness_i}{fitness_{max}} + 0.1$

- Step 8. Generate new locations for the all the group members, selected by p_i , by using self experience, global leader experience and group member's experiences.

$$SM_{newij} = SM_{ij} + \phi_1 \times (GL_j - SM_{ij}) + \phi_2 \times (SM_{rj} - SM_{ij}) + \phi_3 (SM_{rj} - SM_{ij}) \text{ Where } \phi_1 \in (0,1) \phi_2 \in (-1,1), \phi_3 \text{ decided by GSS}$$

- Step 9. Update the position of local and global leaders, by applying the greedy selection process on all the groups.

- Step 10. If any Local group leader is not updating her position after a specified number of times (LL_{Limit}) then re-direct all members of that particular group for foraging by algorithm

if $U(0,1) \geq pr$

$$SM_{newij} = SM_{minj} + \phi \times (SM_{maxj} - SM_{minj}) \text{ Where } \phi \in (0,1)$$

else

$$SM_{newij} = SM_{ij} + \phi \times (GL_j - SM_{ij}) + \phi \times (SM_{ij} - LL_{kj}) \text{ Where } \phi \in (0,1)$$

- Step 11. If Global Leader is not updating her position for a specified number of times (GL_{Limit}) then she divides the group into smaller groups by following steps.

if Global Limit Count > GL_{Limit} then set Global Limit Count = 0

if Number of groups < MG then

Divide the population into groups.

else

Combine all the groups to make a single group.

Update Local Leaders position.

IV. Experimental Results

A. Test Problems

In order to check performance of modified position update in SMO algorithm it is tested over some well known benchmark optimization functions f_1 to f_9 (Listed in Table I). These are continuous optimization problems and have different degrees of complexity, search range and multimodality. Test problems are taken from [16], [17] with the associated offset values.

B. Experimental setting

To prove the competence of MPU-SMO, it is compared with original SMO algorithms. To test MPU-SMO over considered problems, subsequent experimental setting is adopted [10]:

- The Swarm size $N = 50$,
 - $MG = 5$,
 - Global Leader Limit (GL_{limit})=50,
 - Local Leader Limit (LL_{limit})=1500,
 - $pr \in [0.1, 0.4]$, linearly increasing over iterations,
- All other parameter setting remains as in its original SMO algorithm [10].

C. Experimental Result Comparison

Statistical results of MPU-SMO with experimental setting as per last subsection are outlined in Table II. Table II show the comparison of results based on mean function value (MFV), standard deviation (SD), mean error (ME), average function evaluations (AFE) and success rate (SR) are accounted. Table II shows that most of the time MPU-SMO do better than other considered algorithms in terms of competence (with less number of function appraisals) and correctness. Table III shows upshots of table II between MPU-SMO and original SMO algorithm. The proposed algorithm at all times gets better AFE and most of the time it also improve SD and ME. It is due to newly introduced modified GSS process in local leader phase and global leader phase.

Table I Test problems

Test Problem	Objective Function	Search Range	Optimum Value	D	Acceptable Error
Rosenbrock	$f_1(x) = \sum_{i=1}^{D-1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2$	[-30, 30]	$f(0) = 0$	30	1.0E-01
Zakharov	$f_2(x) = \sum_{i=1}^D x_i^2 + (\sum_{i=1}^D \frac{ix_i}{2})^2 + (\sum_{i=1}^D \frac{ix_i}{2})^4$	[-5.12, 5.12]	$f(0) = 0$	30	1.0E-02
Inverted Cosine wave function	$f_3(x) = -\sum_{i=1}^{D-1} (\exp(\frac{-(x_i^2 + x_{i+1}^2 + 0.5x_ix_{i+1})}{8}))$ 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 Problem (NF3)	$f_4(x) = \sum_{i=1}^D (x_i - 1)^2 - \sum_{i=2}^D x_ix_{i-1}$	[-100, 100]	$f(0) = -210$	10	1.0E-01
Colville function	$f_5(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$	[-10, 10]	$f(1) = 0$	4	1.0E-05
Braninss Function	$f_6(x) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f) \cos x_1 + e$	$x_1 \in [-5, 10],$ $x_2 \in [0, 15]$	$f(-\pi, 12.275) = 0.3979$	2	1.0E-05
Kowalik function	$f_7(x) = \sum_{i=1}^{11} (a_i - \frac{x_i(b_i^2 + b_ix_2)}{b_i^2 + b_ix_3 + x_4})^2$	[-5, 5]	$f(0.1928, 0.1908, 0.1231, 0.1357) = 3.07E-04$	4	1.0E-05
2D Tripod function	$f_8(x) = p(x_2)(1 + p(x_1)) + (x_1 + 50p(x_2)(1 - 2p(x_1))) + (x_2 + 50(1 - 2p(x_2))) $	[-100, 100]	$f(0, -50)=0$	2	1.0E-04
Shifted Rosenbrock	$f_9(x) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2 + f_{bias})$ $z = x - o + 1, x = [x_1, x_2, ..., x_D], o = [o_1, o_2, ..., o_D]$	[-100, 100]	$f(o) = f_{bias} = 390$	10	1.0E-01

Table II Comparison of the results of test problems

Test Problems	Algorithm	MFV	SD	ME	AFE	SR
f_1	SMO	4.86E+01	4.10E+01	4.86E+01	203129.5	0
	MPU-SMO	7.18E+01	6.17E+01	7.18E+01	200085.7	1
f_2	SMO	2.58E-02	2.03E-02	2.58E-02	198086.6	21
	MPU-SMO	1.16E-02	4.84E-03	1.16E-02	150209.7	82
f_3	SMO	-8.99E+00	5.22E-02	5.61E-03	62104.7	98
	MPU-SMO	-9.00E+00	1.47E-06	8.23E-06	88707.36	100
f_4	SMO	1.54E+02	7.05E+02	1.54E+02	146426.6	62
	MPU-SMO	5.65E+02	1.12E+03	5.65E+02	140628.8	65
f_5	SMO	2.53E-05	8.74E-05	2.53E-05	121470.4	87
	MPU-SMO	7.28E-06	2.23E-06	7.28E-06	110818.1	100
f_6	SMO	3.98E-01	6.84E-06	6.07E-06	32822.14	85
	MPU-SMO	3.98E-01	6.71E-06	5.75E-06	25335.75	89
f_7	SMO	3.29E-04	9.93E-05	2.15E-05	113055.6	95
	MPU-SMO	3.16E-04	1.74E-06	8.19E-06	93956.36	100
f_8	SMO	6.55E-05	2.46E-05	6.55E-05	10498.41	100
	MPU-SMO	6.50E-05	2.77E-05	6.50E-05	5926.14	100
f_9	SMO	3.91E+02	6.84E+00	1.45E+00	129334.4	75
	MPU-SMO	3.90E+02	3.92E-01	1.37E-01	97455.31	93

V. CONCLUSION

This paper suggests two changes in original SMO algorithm. Both local leader and global leader phases are modified by incorporating GSS process. Newly added steps are inspired by memetic search in ABC and position update achieved on the basis of appropriateness of individual in order to balance intensification and diversification of local search breathing space. Additional, the advanced strategy is applied to get to the bottom of 9 well-known benchmark functions. With the help of experiments over test problems, it is shown that the addition of the proposed strategy in the original SMO improves the trustworthiness, competence and accurateness as weigh against to their original adaptation. Table II and III show that the anticipated MPU-SMO is competent to solve largest part the considered problems with smaller amount of time and efforts.

Table III Summary of table II outcomes

Test Problem	MPU-SMO vs. SMO
f_1	+
f_2	+
f_3	+
f_4	+
f_5	+
f_6	+
f_7	+
f_8	+
f_9	+
Total number of + sign	9

VI. References

- [1] XS Yang. Nature-inspired metaheuristic algorithms. Luniver Press, 2011.
- [2] M Dorigo et al. "Ant colony optimization: a new meta-heuristic". In Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress, volume 2. IEEE, 1999.
- [3] J Kennedy et al. "Particle swarm optimization. In Neural Networks, 1995". Proceedings, IEEE International Conference on, volume 4, pages 1942–1948. IEEE, 1995.
- [4] KV Price et al. "Differential evolution: a practical approach to global optimization". Springer Verlag, 2005.
- [5] J Vesterstrom et al. "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems". In Evolutionary Computation, 2004. CEC2004. Congress on, volume 2, pages 1980–1987. IEEE, 2004.
- [6] KM Passino. "Biomimicry of bacterial foraging for distributed optimization and control". Control Systems Magazine, IEEE, 22(3):52–67, 2002.
- [7] D Karaboga. "An idea based on honey bee swarm for numerical optimization". Techn. Rep. TR06, Erciyes Univ. Press, Erciyes, 2005.
- [8] D Karaboga et al. "A comparative study of artificial bee colony algorithm". Applied Mathematics and Computation, 214(1):108–132, 2009.
- [9] G Zhu et al. "Gbest-guided artificial bee colony algorithm for numerical function optimization". Applied Mathematics and Computation, 217(7):3166–3173, 2010.
- [10] JC Bansal, H Sharma, SS Jadon, & M Clerc,(2013). Spider Monkey Optimization algorithm for numerical optimization. Memetic Computing, 1-17.
- [11] E Bonabeau, M Dorigo, and G Theraulaz. Swarm intelligence: from natural to artificial systems. Number 1. Oxford University Press, USA, 1999.
- [12] JC Bansal, H Sharma, KV Arya and A Nagar, "Memetic search in artificial bee colony algorithm." Soft Computing (2013): 1-18.
- [13] S Kumar, VK Sharma and R Kumari (2014) Randomized Memetic Artificial Bee Colony Algorithm. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS).In Print.
- [14] S Kumar, VK Sharma and R Kumari (2014) Memetic search in differential evolution algorithm, International Journal of Computer Application. In print.
- [15] J Kiefer (1953) Sequential minimax search for a maximum. In: Proceedings of American Mathematical Society, vol. 4, pp 502–506.
- [16] MM Ali, C Khompataporn, and ZB Zabinsky. "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems." J. of Global Optimization, 31(4):635–672, 2005.
- [17] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, YP Chen, A. Auger, and S. Tiwari. "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization." In CEC 2005, 2005.