Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 127 – 134

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Centroid Mutation Embedded Shuffled Frog-Leaping Algorithm

Shweta Sharma[a], Tarun K. Sharma[a,*], Millie Pant[b], J. Rajpurohit[a], B. Naruka[a]

[a]*Amity University Rajasthan, Jaipur – 303002, India*
[b] *Indian Institute of Technology Roorkee, Roorkee – 247667, India*

Abstract

Stochastic search algorithms that take their inspiration from nature are gaining a great attention of many researchers to solve high dimension and non – linear complex optimization problems for which traditional methods fails. Shuffled frog – leaping algorithm (SFLA) is recent addition to the family of stochastic search algorithms that take its inspiration from the foraging process of frogs. SFLA has proved its efficacy in solving discrete as well as continuous optimization problems. The present study introduces a modified version of SFLA that uses geometric centroid mutation to enhance the convergence rate. The variant is named as Centroid Mutated – SFLA (CM-SFLA). The proposal is implemented on five benchmark and car side impact problem. Simulated results illustrate the efficacy of the proposal in terms of convergence speed and mean value.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Shuffled frog – leaping algorithm; SFLA; Global optimization; Geometric centroid; Stochastic; car side impact;

1. Introduction

Optimization in simple terms defined as choosing the best alternative from the given set of solutions. Optimization problems exist in almost every sphere of human activities. Optimization techniques are widely used where decisions have to be taken in some or more complex conditions that can be formulated mathematically. To solve such complex high dimension and real world optimization problems, stochastic search techniques gathers the attention of many researchers, scientists and academicians. Stochastic search techniques or nature inspired metaheuristic algorithms (NIMA) mimic their inspiration from nature or some biological phenomenon. Some of the

* Corresponding author. Tel.: +91-9456086759.
E-mail address: tanu1dpt@iitr.ernet.in

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.003

popular stochastic search techniques are GA (Genetic Algorithm)[1], DE (Differential Evolution)[2] given by Price and Storn, PSO (Particle Swarm Optimization)[3] introduced by Kennedy and Eberhart in 1995, ABC (Artificial Bee Colony)[4] conceptualized by Karaboga in 2005, SFLA[5] introduced by Eusuff and Lansey in 2003 etc. Application of these algorithms in solving intricate and complex optimization problems emerging in various real worlds has proved their efficacy.

SFLA is a recent addition to the family of stochastic search methods that mimics the social and natural behavior of species. SFLA is formulated on the concept of evolution of memeplexes in Frogs. SFLA combines the advantages of local search process of particle swarm optimization (PSO) and information exchanging of the shuffled complex evolution. The basic idea behind modeling of such algorithms is to achieve near to global solutions to the large scale optimization problems and complex problems which can't be solved using deterministic or traditional numerical techniques. SFLA has also proved its efficacy and ability in discovering global optimal solutions to several combinatorial optimization problems[5]. In this study we have incorporated geometric centroid mutation operator to enhance the convergence rate of basic SFLA. The resulting algorithm is named as Centroid Mutated – SFLA (CM-SFLA).

The paper is organized as follows: Basic SFLA is given in Section 2, followed by Section 3, which describes the proposed CM-SFLA and problem definitions are given in Section 4. The simulation strategy with results is discussed in Section 5. Finally, the paper concludes with Section 6.

2. Outline of SFLA

SFLA, stochastic search algorithm based on evolution of memeplexes. In essence, SFLA contains the element of both the local search method of PSO (particle swarm optimization) and the concept of mixing information of the shuffled complex evolution. Since inception SFLA has proved its efficacy and has been applied successfully in finding global solutions to several real world global optimization problems [6,7]. In SFLA, a set of frogs represents the population of possible solutions, which is partitioned into subsets called memeplexes. Different subsets are having frogs from different culture and each frog carry out a local search and the position of worst's frog is modified or updated so that the frogs can move towards optimization. When each subset evolves through fixed number of generations or memetic evolution steps, the ideas hold by the frogs within the subset are passed among subsets through shuffling process. This process of local search and shuffling of information continues until the termination criterion is satisfied.

There are four steps in SFLA:

Initialization Process

The population of frogs P_F is generated randomly. The frog positions (solutions) are given by

$$X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,S}) \tag{1}$$

where S, denotes the dimension.

Sorting and Division Process

The frogs, based on their fitness evaluations are sorted in descending order. Then the sorted population of P_F frogs is distributed into m subsets (memeplexes) each subset holds n frogs such that $P_F = m \times n$. The distribution is done such that the frog with maximum fitness value will go into subset first, accordingly the next frog into second subset and so on. Then X_b (best) and $X_{w,\ (new)}$ (worst) individuals in each subset are determined.

Local Search Process

Worst individual position is improved using equation (2) and (3):

$$D_i = rand(0,1) \times (X_b - X_w) \tag{2}$$

$$X_{w,(new)} = X_w + D_i;\ -D_{\max} \le D_i \le D_{\max} \tag{3}$$

where i = 1, 2, …, N_{gen}; D is the movement of a frog whereas D_{max} represents the maximum permissible movement of a frog in feasible domain; N_{gen} is maximum generation of evolution in each subset. The old frog is replaced if the evolution produces the better solution else X_b is replaced by X_g (optimal solution). If no improvement is observed then a random frog is generated and replaces the old frog. This process of evolution continues till the termination criterion met.

Shuffling Process

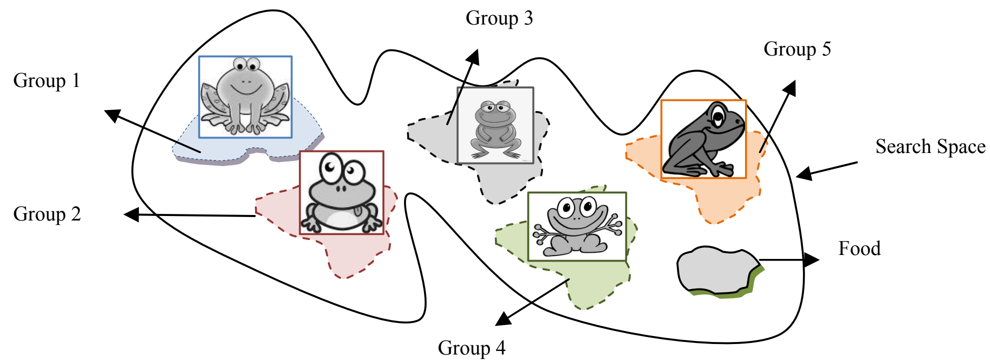
The frogs are again shuffled and sorted to complete the round of evolution. Again follow the same four steps until the termination condition met. The pictorial representation of frog in SFLA is shown in Fig. 1(a).

3. Centroid Mutated SFLA (CM-SFLA)

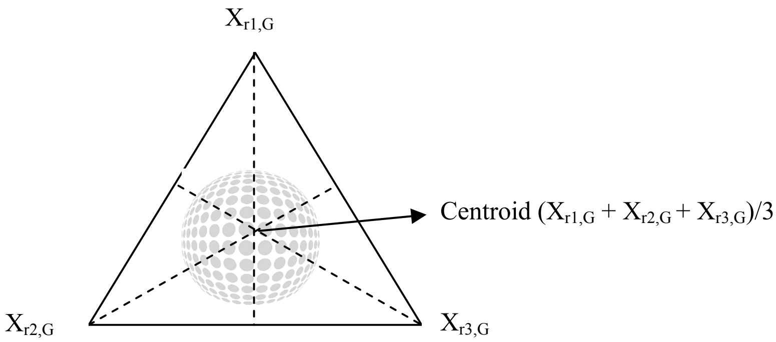
SFLA, despite having prominent features, it is sometimes criticized in terms of convergence rate and getting trapped in local optima for some computationally expensive functions. The paper proposal introduce and incorporate geometric centroid mutation (GCM) operator[8] with a probability based mutation parameter named as C_m to enhance the convergence rate of basic SFLA. GCM operator is stochastically applied, depending on the probability (C_m) in CM-SFLA. In each generation, if the randomly generated number between 0 and 1 is less than or equals to C_m then the new frog position is calculated according to the equation (4) otherwise the frog position is updated using equation (2). Mathematically GCM expression is as follows:

$$D_{i,G} = (X_{\min,G} + X_{r_1,G} + X_{r_2,G})/3) + rand(0,1) \times (X_{b,G} - X_{w,G}) \tag{4}$$

where $X_{r1,G}$ $X_{r2,G}$ are random frogs distinct and different from best and current frogs, $X_{min,G}$ is the frog's best position based on fitness function value. The pictorial representation of geometric centroid mutation is shown in Fig. 1(b).



(a)



(b)

Fig. 1(a). Group searching, initially locally then exchanging information with other groups, for food search in SFLA. 1(b). Geometric Centroid formed with the help of three random chosen points.

Computational steps of CM-SFL algorithm

Begin;
Generate random population of P_F solutions (frogs);
For each frog $i \in P_F$: calculate fitness (i);
    Sort the Frog population P_F in descending order of their fitness;
    Divide P_F into m Memeplexes;
For each Memeplexes;
    Position of best and worst frogs is determined;
    Define GCM probability C_m and select three random positions from population of frogs and generate new position of frog (a) using GCM operator, equation (4) with probability C_m; (b) using equation (3) with probability $(1 – C_m)$;
Repeat for a fixed number of iterations;
End;
    Combine the evolved Memeplexes;
    Sort the Frog population P_F in descending order of their fitness;
    Check if termination criterion = true;
End;

4. Test Bed

The efficiency of proposed CM-SFLA is tested on a set of:

- Five Benchmark Problems[9]:

a) *Sphere*: $F_1(k) = \sum_{i=1}^{D} k_i^2$; Search Range: $(-100, 100)^D$; theoretical Optimum: $F_{min} = 0$.

b) *Rosenbrock*: $F_2(k) = \sum_{i=1}^{D-1} [100(k_{i+1} - k_i^2)^2 + (k_i - 1)^2]$; Search Range: $(-30, 30)^D$; theoretical Optimum: $F_{min} = 0$.

c) *Rastrigin*: $F_3(k) = \sum_{i=1}^{D} [k_i^2 - 10\cos(2\pi k_i) + 10]$; Search Range: $(-5.12, 5.12)^D$; theoretical Optimum: $F_{min} = 0$.

d) *Grienwank*: $F_4(k) = \frac{1}{4000}\sum_{i=1}^{D} k_i^2 - \prod_{i=1}^{D}\cos\left(\frac{k_i}{\sqrt{i}}\right) + 1$; Search Range: $(-600, 600)^D$; theoretical Optimum: $F_{min} = 0$.

e) Ackley: $F_5(k) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} k_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi k_i)\right) + 20 + e$; Search Range: $(-30, 30)^D$; theoretical Optimum: $F_{min} = 0$.

- Car Side Impact Problem: The cars have to pass certain standard tests to measure their safety measures. One of the safety measures is effect of side impact that is measured on a dummy. Further details can be found in Gu et al.[10]. The pictorial representation is shown in Fig. 1. The problem is to minimize the weight and mathematically formulated as:

$Minimize\ \ f(x) = Weight$

subject to:

$g_1(k) = F_a(load\ in\ abdomen) \le 1\ kN$; $g_2(k) = VC_u(dummy\ upper\ chest) \le 0:32\, m/s$

$g_3(k) = VC_m(dummy\ \ middle\ \ chest) \le 0:32\,m/s\ \ g_4(k) = VC_l(dummy\ \ lower\ \ chest) \le 0:32\,m/s$
$g_5(k) = \Delta_{ur}(upper\,rib\,deflection) \le 32\,mm\ \ g_6(k) = \Delta_{mr}(middle\,rib\,deflection) \le 32\,mm$
$g_7(k) = \Delta_k(lower\,rib\,deflection) \le 32\,mm\ \ g_8(k) = F_p(Public\ \ force) \le 4\,kN$
$g_9(k) = V_{MBP}(Velocity\,of\,V - Pillar\,at\,middle\,po\text{int}) \le 9:9\,mm/ms$
$g_{10}(k) = V_{FD}(Velocity\,of\ \ front\,door\,at\ \ V - Pillar) \le 15:7\,mm/ms$

Simplified model is presented as:
$Weight = 1.98 + 4.90k_1 + 6.67k_2 + 6.98k_3 + 4.01k_4 + 1.78k_5 + 2.73k_7$
$F_a = 1.16 - 0.3717k_2k_4 - 0.00931k_2k_{10} - 0.484k_3k_9 + 0.01343k_6k_{10}$
$VC_u = 0.261 - 0.0159k_1k_8 - 0.019k_2k_7 + 0.0144k_3k_5 + 0.0008757k_5k_{10} + 0.08045k_6k_9 + 0.00139k_8k_{11} + 0.00001575k_{10}k_{11}$
$VC_m = 0.214 + 0.00817\ k_5 - 0.131\ k_1k_8 - 0.0704\ k_1k_9 + 0.03099\ k_2k_6 - 0.018\ k_2k_7 + 0.0208\ k_3k_8 + 0.121\ k_3k_9 - 0.00364\ k_5k_6 + 0.0007715\ k_5k_{10} - 0.0005354\ k_6k_{10} + 0.00121\ k_8k_{11}$
$VC_l = 0.74 - 0.061k_2 - 0.163k_3k_8 + 0.001232k_3k_{10} - 0.166k_7k_9 + 0.227k_2^2$
$\Delta_{ur} = 28.98 + 3.818k_3 - 4.2k_1k_2 + 0.0207k_5k_{10} + 6.63k_6k_9 - 7.7k_7k_8 + 0.32k_9k_{10}$
$\Delta_{mr} = 33.86 + 2.95k_3 + 0.1792k_{10} - 5.057k_1k_2 - 11.0k_2k_8 - 0.0215k_5k_{10} - 9.98k_7k_8 + 22.0k_8k_9$
$\Delta_{lr} = 46.36 - 9.9k_2 - 12.9k_1k_8 + 0.1107k_3k_{10}$
$F_p = 4.72 - 0.5k_4 - 0.19k_2k_3 - 0.0122k_4k_{10} + 0.009325k_6k_{10} + 0.000191k_{11}^2$
$V_{MBP} = 10.58 - 0.674k_1k_2 - 1.95k_2k_8 + 0.02054k_3k_{10} - 0.0198k_4k_{10} + 0.028k_6k_{10}$
$V_{FD} = 16.45 - 0.489k_3k_7 - 0.843k_5k_6 + 0.0432k_9k_{10} - 0.0556k_9k_{11} - 0.000786k_{11}^2$
where $0.5 \le k_1, k_3; k_4 \le 1.5$; $0.45 \le k_2 \le 1.35$; $0.875 \le k_5 \le 2.625$; $0.4 \le k_6, k_7 \le 1.2$; $k_8, k_9 \in \{0.192, 0.345\}$; $0.5 \le k_{10}, k_{11} \le 1.5$.

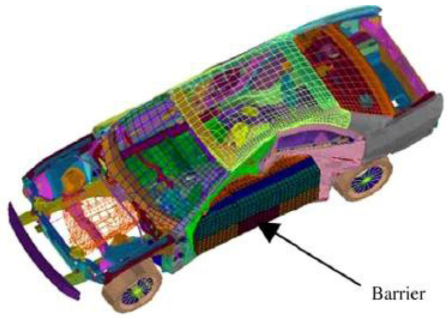


Fig. 2. Car Side impact environment

5. Simulation Strategy

The focus of the present study is on comparing the quality of the simulated results and the number of function evaluations (to analyze the convergence speed) taken to achieve them. For the same average (mean), the standard deviations and the number of function evaluations of the best run were noted. The proposed algorithm CM-SFLA is compared with other algorithms like DE (differential evolution), PSO (particle swarm optimization), ABC (artificial bee colony)[9] and SFLA (shuffled frog leaping algorithm).

The proposed CM-SFLA is executed in Dev C++.

6. Parameterization and Result Analyses

The parameter settings of SFLA and CM-SFLA, for the fair comparison are stated in Table 1. The population of frogs is generated using inbuilt *rand*() function. The statistical results of the proposal on five benchmark problems are compared and analyzed with the simulation results of DE, PSO, ABC & SFLA and for car side impact problem the simulated results are compared with DE, PSO, GA, ABC & SFLA for a valid comparison.

Table 1. Parameterizations for test systems.

Population Size of Frogs	200
GCM probability	0.5
Memeplexes (*m*)	20
Local Explorations iterations in each Memeplexes	10
Number of Function Evaluations (NFE)	50000
D_{max}	100% of variable range
Constrained Handling	Pareto Front Method

Result Analysis:

- **Benchmark Problems:**

The results of five benchmark problems, obtained from simulation are presented in Table 2, 3, and 4. Table 2 presents the simulated results in terms of mean fitness value and standard deviation (Std. Dev.) whereas the total NFE (number of function evaluation) taken by each problem to achieve optimal result in each case are given in Table 3. From the results it is clear that for all the problems, the CM-SFLA performed well in terms of mean values and comparatively taken fewer number of NFE's to achieve them. The Acceleration rate achieved by the CM-SFLA with respect to DE, PSO, ABC and SFLA is shown in Fig. 3(b). The result in terms of NFE justifies the convergence speed of the proposal.

A further statistical analysis[11,12] is performed to test the efficiency of the algorithms. To detect significant difference for the CM-SFLA algorithm, Bonferroni–Dunn test[13] is used to perform Post-hoc test.

Bonferroni–Dunn's graph, to examine significant difference between algorithms, for all the test problems is shown in Fig 3(a). A horizontal line is drawn to show two levels of significance, $\alpha = 0.05$ and $\alpha = 0.10$. The formula used to calculate critical difference (CD) is given as:

$$CD = Q_\alpha \sqrt{\frac{a(a+1)}{6N}}$$

where Q_α is the critical value for a multiple non-parametric comparison with a control[14], a is the number of algorithms and the number of the problems taken for comparison are symbolized by N.

Bonferroni-Dunn's test notifies the subsequent significant differences with:

- *CM-SFLA as control algorithm:*

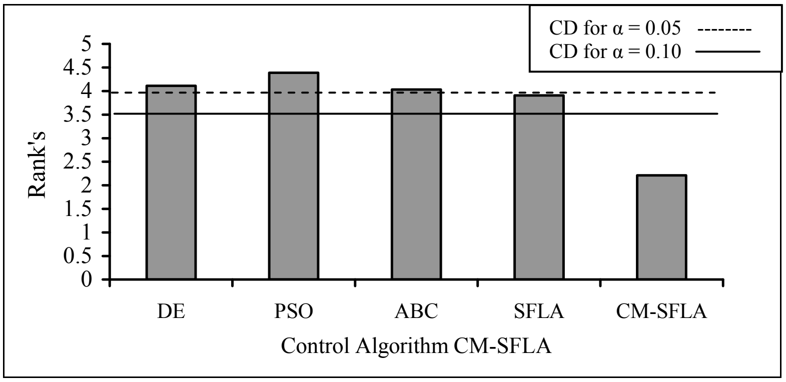
CM-SFLA is better than DE, PSO, ABC and SFLA at α = 0.05 and α = 0.10.

Table 2. Simulated results of benchmark test systems.

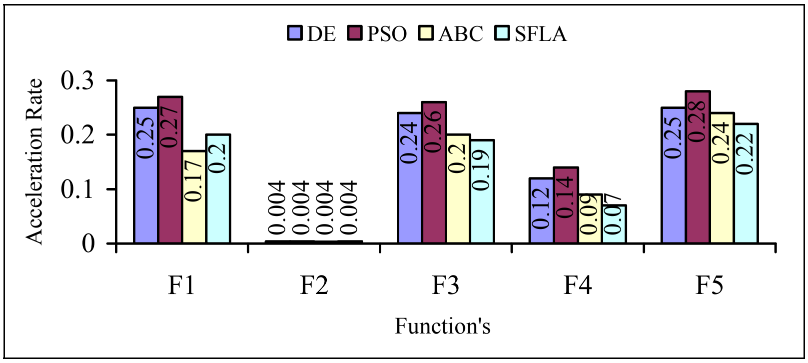
Algorithm	Statistics	F_1	F_2	F_3	F_4	F_5
DE	Mean	1.4635E-17	3.4618E-03	9.7532E-15	1.7267E-17	3.2272E-14
	±Std. Dev.	2.3291E-18	±1.9266E-03	±2.9273E-15	±3.7219E-18	±2.1945E-12
PSO	Mean	4.1761E-16	2.6591E-03	1.3263E-14	2.9606E-17	3.2152E-14
	±Std. Dev.	±7.365E-17	±2.220E-03	±2.445E-14	±4.993E-17	±3.252E-15
ABC	Mean	6.264E-16	5.99147E-02	1.6985E-14	1.0954E-14	2.03741E-13
	±Std. Dev.	±1.2025E-16	±3.2512E-02	±6.9865E-14	±1.0875E-15	±6.1286E-15
SFLA	Mean	2.0535E-16	1.2362E-01	2.6252E-14	2.0162E-14	2.1022E-13
	±Std. Dev.	±3.1731E-16	±5.0001E-02	±2.8732E-14	±2.2272E-16	±4.8367E-15
CM-SFLA	Mean	1.2613E-18	1.3261E-04	3.7725E-15	1.2726E-17	3.0921E-15
	±Std. Dev	±2.6001E-19	±4.1233E-05	±5.9272E-17	±2.3947E-19	±1.1038E-16

Table 3. NFE taken by bench mark test systems .

Functions	DE	PSO	ABC	SFLA	CM-SFLA
F_1	31765	32974	28848	29944	23973
F_2	399990	399989	399750	399867	398360
F_3	75283	76491	71083	70635	56932
F_4	44932	45973	43830	42673	39755
F_5	56011	57923	54987	53837	41873



(a)



(b)

Fig. 3(a) Bonferroni-Dunn's graphic corresponding to error. (b) Acceleration Rate of CM-SFLA with respect to DE, PSO, ABC, SFLA

Table 4. Ranking and critical difference calculated through Friedman's and Bonnferroni-dunn's procedure

Algorithm	Mean Rank
DE	4.11
PSO	4.39
ABC	4.03
SFLA	3.91
CM-SFLA	2.21
CD for α = 0 .05	3.678
CD for α = 0 .10	3.476

- **Car Side Impact**

The comparative statistical results obtained after 20,000 searches from simulation of SFLA & CM-SFLA and using other stochastic techniques (taken from literature) such as DE, PSO, GA, FA and ABC are summarized in Table 5. During the simulation, uniform settings are maintained for fair comparison as the performance may differ in terms of computational time. The best function value achieved by the proposal is comparatively better[15] than PSO, DE, GA, ABC and SFLA. Further, only PSO shows smaller Std. Dev.

Table 5. Optimized results of Car side impact problem

Algorithms	PSO	DE	GA	FA	ABC	SFLA	CM-SFLA
Best Function Value	22.84474	22.84298	22.85653	22.84298	22.84839	22.8363	22.84299
k_1	0.50000	0.50000	0.50005	0.50000	0.5	0.5	0.5
k_2	1.11670	1.11670	1.28017	1.36000	1.183	1.172	1.1121
k_3	0.50000	0.5000	0.50001	0.50000	0.50001	0.5	0.5
k_4	1.30208	1.30208	1.03302	1.20200	1.202	1.382	1.201
k_5	0.50000	0.50000	0.50001	0.50000	0.5001	0.5	0.5
k_6	1.50000	1.50000	0.50000	1.12000	1.12	1.13	1.51
k_7	0.50000	0.50000	0.50000	0.50000	0.5000	0.5	0.5
k_8	0.34500	0.34500	0.34994	0.34500	0.34491	0.34456	0.35001
k_9	0.19200	0.19200	0.19200	0.19200	0.192	0.1892	0.19011
k_{10}	-19.54935	-19.54935	10.3119	8.87307	8.87295	11.082	10.0982
k_{11}	-0.00431	-0.00431	0.00167	-18.99808	-18.99749	-0.0921	-9.1862
Mean Function Value	22.89429	23.22828	23.51585	22.89376	22.8857	23.1033	22.8679
Worst Function Value	23.21354	24.12606	26.240578	24.06623	24.8193	26.0834	24.0101
Std. Dev.	0.15017	0.34451	0.66555	0.16667	0.17393	0.36537	0.15171

7. Conclusions

The proposed study suggests a simple but efficient modification in the structure of SFLA by introducing the geometric centroid mutation to enhance the convergence rate. The statistical results on benchmark problems show that the proposed modification significantly helps in improving the performance of basic SFLA. The results are also validated statistically for the proposed CM-SFLA, where once again it was shown significantly better than other algorithms. Further the efficacy of the proposal is tested on real world problem of car side impact. The simulated result shows that the proposal is capable of solving constrained real world problems of optimization.

In future we will try to implement it to multi objective problems with some more modifications.

References

1. Goldberg D. *Genetic Algorithms in Search, Optimization, and Machine Learning*. MA:Addison Wesley;1989.
2. Price K, Storn R. *Differential Evolution – a Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces, Technical Report*. Berkley: International Computer Science Institute;1995.
3. Kennedy J, Eberhart RC. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks,Perth, Australia, IEEE Service Center, Piscataway, NJ* 1995:1942–1948.
4. Karaboga D. *An Idea based on Bee Swarm for Numerical Optimization,Technical Report, TR-06*. Erciyes University Engineering Faculty, Computer Engineering Department ;2005.
5. Eusuff M, Lansey KE. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Water Resources Planning and Management* 2003;129(3):210–225.
6. Kumar JV, Kumar VDM. Generation bidding strategy in a pool based electricity market using Shuffled Frog Leaping Algorithm. *Applied Soft Computing* 2014;21:407–414.
7. Arandian B, Hooshmand RA, Gholipour E. Decreasing activity cost of a distribution system company by reconfiguration and power generation control of DGs based on shuffled frog leaping algorithm. *International Journal of Electrical Power & Energy Systems* 2014;61:48–55.
8. Ali M, Pant M, Nagar A. Two new approach incorporating centroid based mutation operators for Differential Evolution. *World Journal of Modelling and Simulation* 2011;7(1):16–28.
9. Sharma TK, Pant M, Ahn CW. Improved Food Sources in Artificial Bee Colony. *Proceeding of IEEE SSCI* 2013:95-102.
10. Gu L, Yang RJ, Cho CH, Makowski M, Faruque M, Li Y. Optimization and robustness for crashworthiness. *Int J Vehicle Des* 2001;26(4):348–60.
11. Demšar J. Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 2006; 7: 1–30.
12. García S, Herrera F. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *J Mach Learn Res* 2008;9:2677–2694.
13. Dunn OJ. Multiple comparisons among means. *J Am Stat Assoc* 1961;56(293):52–64.
14. Zar JH. *Biostatistical analysis*. Englewood Cliffs :Prentice-Hall;1999.
15. Sharma TK, Pant M. Modified Onlooker Phase in Artificial Bee Colony Algorithm. *Proceedings Swarm, Evolutionary, and Memetic Computing, Lecture Notes in Computer Science, Springer Berlin Heidelberg* 2012;7677:339-347.