Disruption Operator-based Spider Monkey Optimization Algorithm

Avinash Kaur[a], Harish Sharma[b] and Nirmala Sharma[c]
Rajasthan Technical University, Kota, India
Email: [a]kauravinash05@gmail.com, [b]harish.sharma0107@gmail.com, [c]nsharma@rtu.ac.in

***Abstract*—Spider monkey optimization (SMO) algorithm, is a neoteric Swarm Intelligence based algorithm, inducemented from the social comportment of spider monkeys, commonly known as fission-fusion social system. Despite of perpetuating an equilibrium state betwixt intensification and diversification by its own, SMO sometimes converges at a particular point due to its swarming nature. To overcome this problem, a new stage, namely disruption stage is incorporated with SMO. The proffered variant is named as Disruption operator-based spider monkey optimization ($DiSMO$) algorithm. In the incorporated stage, the disruption operator helps to scatter the swarms in convergence condition and according to the difference from the best solution, this operator is employed to entire solutions so that the solutions might attract or distract from the best solution. Further, the efficiency of the proffered strategy is estimated over 12 different benchmark functions and the outputs are being compared with basic SMO, its significant variant: power law based local search in SMO ($PLSMO$), and particle swarm optimization (PSO).**

***Index Terms*—Meta-heuristic Optimization Techniques; Spider monkey optimization algorithm; Swarm Intelligence; Disruption Operator**

I. Introduction

Some complex problems are very arduous to solve through existing numerical methods but swarm intelligence based algorithms use tentative methods and not only learn from individuals behavior but also from its neighbour [5]. Spider monkey optimization (SMO) algorithm is a widespread swarm intelligence algorithm which uses the stochastic search study i.e. works on previous intelligence and randomness. This algorithm is developed by conceiving motivation from the cognitive behavior and the proficiency of communication of spider monkeys [2]. But it is always probable that SMO erratically stops emanating towards global optima and gets stuck around local optimum [2, 9]. Researchers are ceaselessly slogging to rectify the potential of the SMO algorithm. [3, 4, 7, 8, 9, 10, 12] Taking the idea from astrophysics, disruption operator [3, 11], which is a new operator, is implemented over SMO for enhancing the exploration and exploitation flair of SMO algorithm. Astrophysics says that, whenever a species of gravitationally restricted particles verge on a huge body M, the species of particles tend to be scattered. The similar process is implemented over the basic SMO. In this, the ideal solution (having the best fitness value) is selected as a lead solution and remaining solutions disrupt due to the influence of gravity force of the lead solution. The proffered algorithm is termed as Disruption operator-based spider monkey optimization ($DiSMO$) algorithm. Remaining paper is categorized accordingly: Section II describes SMO algorithm. In Section III, proffered disruption phenomenon in SMO is explained. A similitude disquisition is demonstrated in Section IV. Finally, Section V encompasses the surmised work.

II. Spider Monkey Optimization (SMO) Algorithm

The algorithm contains six stages i.e. local leader stage, global leader stage, global leader learning stage, local leader learning stage, local leader decision stage and global leader decision stage [2]. Before this, the populace is initialized and six stages are executed one by one as shown below: Initialization stage: Firstly, SMO produces an evenly scattered initial populace of N solutions where each solution SM_i (i= 1, 2, . . . ; N) is a D-dimensional vector. Here D represents the number of variables in the optimization problem and SM_i is the i^{th} spider monkey in the populace. Each populace is generated by the following equation:-

$$SM_{ij} = SM_{minj} + U(0,1)(SM_{maxj} - SM_{minj}) \quad (1)$$

where, U(0,1) is an evenly scattered random number in the range [0,1] and x_{minj} and x_{maxj} are the bounds of x_i in j^{th} direction.

A. Local leader stage:

In this stage, the spider monkeys means the solutions update their positions by taking the experience from local leader as well as from the randomly selected monkey of the whole populace. For i^{th} candidate, the position update equation in this stage is

$$SMnew_{ij} = SM_{ij} + U(0,1)(LL_{kj} - SM_{ij}) + U(-1,1)(SM_{rj} - SM_{ij}) \quad (2)$$

where, $SMnew_{ij}$ is the new position of solution, SM_{ij} is the i^{th} solution of j^{th} dimension and $j \in \{1,2,...,D\}$. LL_{kj} is the local leader in j^{th} dimension of k^{th} group and SM_{rj} is the randomly selected spider monkey from the populace.

B. Global leader stage:

In this stage, the spider monkeys means the solutions update their position by taking the experience from global leader as well as from the randomly selected monkey of the whole

ISBN: 978-1-5090-6471-7/17/$31.00 ©2017 IEEE

populace. For i^{th} candidate, the equation of position update in this stage is:

$$\begin{aligned} SMnew_{ij} = SM_{ij} + U(0,1)(GL_j - SM_{ij}) + \\ U(-1,1)(SM_{rj} - SM_{ij}) \end{aligned} \quad (3)$$

where, $SMnew_{ij}$ is the new position of solution, SM_{ij} is the i^{th} solution of j^{th} dimension and $j \in \{1, 2, ..., D\}$. GL_{kj} represents the global leader in j^{th} dimension and SM_{rj} is the randomly selected spider monkey from the populace.

C. Global leader learning stage:

In this stage, that spider monkey is selected as the global leader which is having the best fitness value in the swarm. Also the position of global leader is checked in each iteration and if it is not updating, then the associated counter namely $GlobalLimitCount$, is incremented by one.

D. Local leader learning stage:

In this stage, that spider monkey is opted as the local leader which is having the best fitness value in the local group. Also the position of local leader is inspected in each iteration and if it is not updating, then the associated counter namely $LocalLimitCount$, is incremented by one.

E. Local leader decision stage:

In this stage, if the $LocalLimitCount$ crosses the predefined $LocalLeaderLimit$, then the local leader is considered as stucked in local optima. Hence, all the local group members are updated either by random initialization or by the following equation as per the perturbation rate (pr):

$$\begin{aligned} SMnew_{ij} = SM_{ij} + U(0,1)(GL_j - SM_{ij}) + \\ U(0,1)(SM_{ij} - LL_{ij}) \end{aligned} \quad (4)$$

It is understood from equation4 that the spider monkeys are moving towards the Global leader and moving away from the local leader so that they can come out from the local value.

F. Global leader decision stage:

In this stage, the counter namely $GlobalLimitCount$ is determined for a particular threshold value which is called, $GlobalLeaderLimit$. If the counter crosses this threshold value, the global leader divides its members into local groups. However this group dividing process is regulated by Maximum group (MG), which is calculated by formula MG=N/10.

G. Spider Monkey Optimization (SMO) Algorithm

SMO algorithm's pseudo-code is as follows: Here, $pr\ \varepsilon$ [0.1,0.8], and N is the swarm size.

Algorithm 1 Spider Monkey Optimization (SMO)

Initialize the populace of solutions, SM_i where (i=1,2,.....,N) using equation 1, $LocalLeaderLimit$, $GlobalLeaderLimit$, pr.
Calculate fitness of the solutions(i.e. the position of food source from spider monkeys)
Select the global leader and local leader.
while if cessation condition is not fulfilled **do**
  Step 1: Produce new positions of solutions $SMnew_{ij}$ by taking the experiences from local leader and by their group members.
  Step 2: Select the solutions by employing the greedy selection process according to their fitness.
  Step 3: Now enumerate the probability ($prob_i$) for all the solutions of the group.
  Step 4: Again update the positions of the solutions according to selected $prob_i$, and experiences of global leader, group members and self-experience.
  Step 5: Update the global leader and local leader again by implementing the greedy selection process on all the groups.
  Step 6: If there is any local leader which is not upgrading its position for a particular number of times ($LocalLeaderLimit$), then the local leader randomly scatters monkeys in search space into different directions in order to forage the food.
  Step 7: Similarly, if global leader is unable to update its position for a particular number of times ($GlobalLeaderLimit$), then she splits the whole group into teeny groups but that should be regulated by MG
end while
Result the ideal solution detected so far.

III. Disruption phenomenon in spider monkey optimization algorithm

When a collection of different particles whose total mass is m, are gravitationally bounded with each other, comes closer to a huge body M, the particles split apart. This is the phenomenon of disruption in nature [6]. The same behaviour is shown in solids too. The occurrence of this phenomenon is caused by the failure of all the forces, supplying influencing pressure to countervail the effects of gravity and retaining the huge body in a state of an equilibrium.

For attaining the proficiency in the optimization algorithms, there should be a suitable balance betwixt exploration and exploitation in the search space. In order to improve this balance, the disruption phenomenon is incorporated as a new stage in the SMO Algorithm.

The proffered stage works by exploring the search space initially for good solutions and then exploiting the search space in the later period. This proffered algorithm is named as $DiSMO$. In $DiSMO$, it is presumed that the solution with the highest fitness value is the 'lead' and under the influence of gravity force of that lead solution, left over solutions are distributed in the search space. The Disruption stage is further narrated as follows[11]:

Disruption condition for the entire candidate solutions (excluding the lead solution), is monitored by:

$$\frac{R_{i,j}}{R_{i,best}} < C \quad (5)$$

here, $R_{i,j}$ portrays the Euclidean distance betwixt the i^{th} and j^{th} candidate solutions. Similarly, Euclidean distance betwixt

i^{th} and the best solution is depicted as $R_{i,best}$. Here, j is the most bordering neighbor of i and C is a threshold which is determined as:

$$C = C_0(1 - \frac{t}{T}) \quad (6)$$

where, t and T indicate the current and maximum number of iterations respectively. The solutions that fulfill the condition of the equation 5 are disrupted due to the proximity to the lead solution. The threshold C is a variable that makes the operator worthwhile. The value of C is used to regulate the exploration and exploitation capabilities: the high value of C is used to explore the search space, while low value will enhance the exploitation capability. The candidate solutions which satisfy equation 5, are updated by the following position update equation.

$$x_i(t+1) = x_i(t) \times \text{D} \quad (7)$$

here, $x_i(t)$ is the position of i^{th} candidate solution at iteration t. Similarly, $x_i(t+1)$ is the position at iteration $t+1$. The estimation of D is translated as:

$$D = \begin{cases} R_{i,j} \times \text{U(-0.5,0.5)}, & \text{if } R_{i,best} >= 1 \\ (1 + \rho \times \text{U(-0.5,0.5)}, & \text{otherwise.} \end{cases} \quad (8)$$

U(-0.5,0.5) shown here, is the uniformly distributed random number in the range [-0.5,0.5] and ρ is a small number, which is utilized to exploit the search space. The exploration and exploitation depends on the value of the D. There are two conditions: first, when $R_{i,best} >= 1$, implies the solutions are not merged and subsequently the value of D can be less than or greater than 1 as shown in equation (8), so D is multiplied with the previous i^{th} value to get the new updated value. If this new updated value is less or more than the preceding solution value, this process performs the exploration. And, second condition is when $R_{i,best} <= 1$, it demonstrates that i^{th} solution is close to the lead solution, consequently the value of D is kept little, so that it can update the position of the solutions nearer to the best solution. This implies, that at whatever point the solutions are assembled or they draw near to the lead solution, the disruption operator exploits the search space otherwise it explores.

Therefore, the proposed $DiSMO$ algorithm contains seven stages, local leader stage, global leader stage, global leader learning stage, local leader learning stage, local leader decision stage, global leader decision stage and disruption stage. The disruption stage is appended after the end of first six stages as shown in the algorithm 2.

Algorithm 2 Disruption operator-based Spider Monkey Optimization Algorithm($DiSMO$)

```
Initialize the parameters: MI (Maximum number of iterations), D (Dimension of the problem), N (Swarm Size), C_0, and ρ;
Initialize the swarms, SM_i where (i=1,2,.....,N) using equation (1)  iter = 1;
while iter <> MI do
  Step 1: Local Leader Stage;
  Step 2: Global Leader Stage;
  Step 3: Global Leader Learning Stage;
  Step 4: Local Leader Learning Stage;
  Step 5: Local Leader Decision Stage;
  Step 6: Global Leader Decision Stage;
  Step 7: Disruption Stage
  for every solution do
    Check the condition utilizing equation (5) for all the candidate solutions barring the lead solution; here, R_{i,j} portrays the Euclidean distance betwixt the i^{th} and j^{th} candidate solutions. Similarly, Euclidean distance betwixt i^{th} and the best solution is depicted as R_{i,best}.;
    C is a threshold ascertained utilizing the equation (6);
    if (R_{i,j}/R_{i,best} < C) then
      D is ascertained utilizing the equation (8);
      Change the position of the solutions utilizing equation (7);
    end if
  end for
  iter=iter+1;
end while
Result the ideal solution detected so far.
```

IV. Experimental results

To scrutinize the interpretation of the proffered $DiSMO$ algorithm, 12 disparate global optimization problems (f_1 to f_{12}) [1, 13] are selected as presented in Table I.

To demonstrate the efficiency of $DiSMO$, a contrastive analysis is taken among $DiSMO$, SMO, its significant variant namely power law based local search in SMO ($PLSMO$), and particle swarm optimization(PSO). The following experimental setting is arrogated to test $DiSMO$, SMO, $PLSMO$, and PSO over the regarded test problems:

- $C_0 = 60$
- $\rho = 10^{-10}$
- The number of simulations/run =100,
- $SwarmsizeN$=50 and $MaximumgroupMG$=N/10,
- Rest of the parameter settings for the algorithms $DiSMO$ and $PLSMO$ are similar to their original research papers.

The showed aftereffects of the regarded algorithms are portrayed in Table II. This Table yields a report that anents the standard deviation (SD), mean error (ME), average number of function evaluations (AFE), and success rate (SR). Results in Table II demonstrate that mostly $DiSMO$ surpasses in terms of reliability, efficiency, and accuracy as compared to the SMO, $PLSMO$, and PSO. Moreover, we measure the average function evaluations ($AFEs$) by contrasting the convergence speed of the examined algorithms. Higher the convergence speed, smaller the AFE.

TABLE I: Test problems

S.No.	Test Problem	Objective function	Search Range	D	Acc Error
1	Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	[-5.12, 5.12]	30	$1.00E-05$
2	De Jong f4	$f_2(x) = \sum_{i=1}^{D} i.(x_i)^4$	[-5.12, 5.12]	30	$1.00E-05$
3	Griewank	$f_3(x) = 1 + \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}})$	[-600, 600]	30	$1.00E-05$
4	Zakharov	$f_4(x) = \sum_{i=1}^{D} {x_i}^2 + (\sum_{i=1}^{D} \frac{ix_i}{2})^2 + (\sum_{i=1}^{D} \frac{ix_1}{2})^4$	[-5.12, 5.12]	30	$1.00E-02$
5	Pathological function	$f_5(x) = \sum_{i=1}^{D-1}\left(0.5 + \frac{sin^2\sqrt{(100x_i^2+x_{i+1}^2)}-0.5}{1+0.001(x_i^2-2x_ix_{i+1}+x_{i+1}^2)^2}\right)$	[-100, 100]	30	$1.00E-05$
6	Sum of different powers	$f_6(x) = \sum_{i=1}^{D} \lvert x_i\rvert^{i+1}$	[-1, 1]	30	$1.00E-05$
7	Step function	$f_7(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5\rfloor)^2$	[-100, 100]	30	$1.00E-05$
8	Inverted cosine wave	$f_8(x) = -\sum_{i=1}^{D-1}\left(\exp\left(\frac{-(x_i^2+x_{i+1}^2+0.5x_ix_{i+1})}{8}\right)\times \mathrm{I}\right)$	[-5, 5]	10	$1.00E-05$
9	Kowalik	$f_9(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$	[-5, 5]	4	$1.00E-04$
10	Shifted Rastrigin	$f_{10}(x) = \sum_{i=1}^{D}(z_i^2 - 10\cos(2\pi z_i) + 10) + f_{bias}$ z=(x-o), x=(x_1,x_2,........x_D), o=(o_1,o_2,........o_D)	[-5, 5]	10	$1.00E-02$
11	Hosaki Problem	$f_{11} = (1 - 8x_1 + 7x_1^2 - 7/3x_1^3 + 1/4x_1^4)x_2^2\exp(-x_2)$, subject to $0 \leq x_1 \leq 5, 0 \leq x_2 \leq 6$	[0, 5] [0, 6]	2	$1.00E-06$
12	Meyer and Roth Problem	$f_{12}(x) = \sum_{i=1}^{5}(\frac{x_1x_3t_i}{1+x_1t_i+x_2v_i} - y_i)^2$	[-10, 10]	3	$1.95E-03$

To limit the stochastic impact of the algorithms, the delineated function evaluations for individual problem is calculated as average over 100 runs. We utilize the acceleration rate (AR) [11] in order to contrast convergence speeds depending on the $AFEs$ which is ascertained as:

$$AR = \frac{AFE_{ALGO}}{AFE_{DiSMO}}, \tag{9}$$

Here, $ALGO \in \{ SMO, PLSMO,$ and $PSO \}$ and $AR > 1$ means $DiSMO$ is speedy. We can audit AR of the proffered algorithm by comparing the basic SMO with its variants, by using equation (9) of AR. Results are analyzed in Table III which depicts a clear comparison betwixt $DiSMO$ and $PLSMO$, $DiSMO$ and SMO, and $DiSMO$ and PSO in terms of AR. So, it is understood that convergence speed of $DiSMO$ is hastier among all the examined algorithms.
Additionally, boxplots evaluation [7] of AFE is performed to contrast all the regarded algorithms on the premise of their amalgamate execution, as it can adequately exhibit the demonstrate the empirical distribution of data pictorially. The boxplots for $DiSMO$, SMO, $PLSMO$, and PSO are portrayed in Figure 1. The outcomes divulges that interquartile range and medians of $DiSMO$ are contrastingly less. Though,

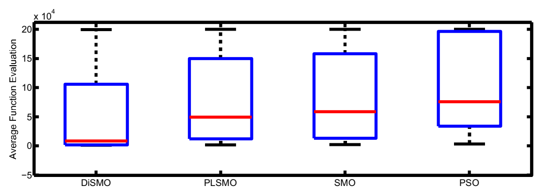

Fig. 1: Boxplots graphs for average number of function evaluation

it is easy to perceive from boxplots that $DiSMO$ is cost efficient than $PLSMO$, SMO, and PSO i.e., $DiSMO's$ outcome differs from other. Now to examine, if there presents any remarkable dissimilarity among algorithm's results or this distinction is because of randomness, we need an alternative statistical test. Boxplots of Figure 1 reveals that the average number of function evaluations utilized by the regarded algorithms to resolve the different problems are not dispersed customarily, so a non-parametric statistical test is expected to separate the accomplishments of the algorithms.
There is a well substantiated, a non-parametric test available namely, Mann-Whitney U rank sum [9], for contrasting among non-Gaussian data. In this paper, this test is conducted at 5% level of consequence ($\alpha = 0.05$) betwixt $DiSMO-PLSMO$, $DiSMO - SMO$, and $DiSMO - PSO$.
Table IV reveals the outcomes for the average function evaluations of 100 simulations of the Mann-Whitney U rank sum test. First we contemplate the consequent dissimilarity by Mann-Whitney U rank sum test i.e., to check whether there is a difference betwixt two data sets significantly or not.

TABLE II: Test problems comparison of the results

Test Problem	Algorithm	SD	ME	AFE	SR
f_1	DiSMO	2.12E-06	8.29E-06	8097.63	100
	PLSMO	8.15E-07	8.96E-06	13174.79	100
	SMO	8.37E-07	8.87E-06	13642.30	100
	PSO	6.10E-07	9.33E-06	38101.50	100
f_2	DiSMO	3.07E-06	2.34E-06	1458.69	100
	PLSMO	9.49E-07	8.68E-06	11198.07	100
	SMO	1.20E-06	8.49E-06	12725.66	100
	PSO	8.62E-07	9.03E-06	32596.50	100
f_3	DiSMO	3.20E-06	6.73E-06	10838.13	100
	PLSMO	4.25E-03	2.13E-03	74710.82	77
	SMO	5.48E-03	2.96E-03	77028.55	72
	PSO	7.12E-03	3.87E-03	113502.50	69
f_4	DiSMO	3.17E-03	4.00E-03	1740.23	100
	PLSMO	4.95E-04	9.65E-03	121187.10	100
	SMO	6.18E-04	9.41E-03	141818.34	100
	PSO	1.80E-02	2.20E-02	196434.00	31
f_5	DiSMO	6.45E-01	9.79E-01	191309.51	5
	PLSMO	8.85E-01	4.25E+00	200000.00	0
	SMO	6.43E-01	3.37E+00	200000.00	0
	PSO	5.20E+00	1.26E-02	200000.00	0
f_6	DiSMO	2.57E-06	6.10E-06	2169.44	100
	PLSMO	1.83E-06	7.99E-06	5288.82	100
	SMO	1.68E-06	7.57E-06	5362.85	100
	PSO	1.70E-06	8.10E-06	9445.00	100
f_7	DiSMO	0.00E+00	0.00E+00	3394.22	100
	PLSMO	0.00E+00	0.00E+00	17812.30	100
	SMO	9.95E-02	1.00E-02	13986.92	99
	PSO	0.00E+00	0.00E+00	37196.00	97
f_8	DiSMO	1.85E-06	7.51E-06	8526.60	100
	PLSMO	1.47E-06	8.41E-06	64565.76	100
	SMO	7.82E-02	7.87E-03	80859.27	97
	PSO	6.84E-01	1.48E+00	195748.00	6
f_9	DiSMO	2.06E-05	8.19E-05	35245.95	100
	PLSMO	1.40E-04	1.19E-04	34295.95	97
	SMO	9.66E-05	1.03E-04	40350.34	97
	PSO	1.26E-05	8.96E-05	35314.50	100
f_{10}	DiSMO	5.01E+00	4.69E+00	199625.27	1
	PLSMO	1.44E+01	1.02E+02	200049.17	0
	SMO	1.40E+01	1.05E+02	200050.56	0
	PSO	4.35E+00	3.60E+01	200050.00	0
f_{11}	DiSMO	3.76E-06	1.03E-05	176171.92	12
	PLSMO	3.63E-06	1.04E-05	178104.34	11
	SMO	3.18E-06	1.07E-05	174092.01	10
	PSO	3.77E-06	1.01E-05	176246.00	10
f_{12}	DiSMO	4.76E-04	1.43E-03	1270.59	100
	PLSMO	3.01E-06	1.95E-03	1755.23	100
	SMO	2.77E-06	1.95E-03	2026.62	100
	PSO	3.01E-06	1.95E-03	3397.00	100

If that distinction is not viewed (i.e., there is null hypothesis) then '=' sign appears otherwise the null hypothesis is rejected and then we can compare the average number of function evaluations. Moreover, we use the signs '+' and '-' for the condition where $DiSMO$ conceives less or more average number of function evaluations, respectively than the other algorithms. That is why, it is shown in Table IV, wherever $DiSMO$ is significantly better, '+' sign appears and '-' sign for viceversa. As Table IV includes 34 '+' signs out of 36 comparisons. Therefore, it can be verdicted that the outcomes of $DiSMO$ is considerably cost effective than $PLSMO$,

TABLE III: Comparison of the basic SMO, $PLSMO$ and PSO based on Acceleration Rate (AR) of $DiSMO$

Test Problems	PLSMO	SMO	PSO
f_1	1.6269933	1.6847275	4.7052656
f_2	7.6767990	8.7240332	22.3464204
f_3	6.8933312	7.1071809	10.4725169
f_4	69.6385535	81.4940209	112.8781828
f_5	1.0454263	1.0454263	1.0454263
f_6	2.4378734	2.4719974	4.3536581
f_7	5.2478331	4.1208054	10.9586297
f_8	7.5722750	9.4831785	22.9573335
f_9	0.9730465	1.1448220	1.0019449
f_{10}	1.0021235	1.0021304	1.0021276
f_{11}	1.0109689	0.9881939	1.0004205
f_{12}	1.3814291	1.5950228	2.6735611

TABLE IV: Depending upon the mean function evaluations and the Mann-Whitney U rank sum test, comparison is done by taking $\alpha = 0.05$.

Test Problems	DiSMO Vs PLSMO	DiSMO Vs SMO	DiSMO Vs PSO
f_1	+	+	+
f_2	+	+	+
f_3	+	+	+
f_4	+	+	+
f_5	+	+	+
f_6	+	+	+
f_7	+	+	+
f_8	+	+	+
f_9	-	+	+
f_{10}	+	+	+
f_{11}	+	-	+
f_{12}	+	+	+

SMO, and PSO over regarded test problems.

V. CONCLUSION

In this paper, to improve diversification ability of SMO algorithm, a new stage namely disruption stage is incorporated with SMO. The proffered variant is named as Disruption operator-based SMO ($DiSMO$). The incorporated stage is based on the disruption phenomenon of astrophysics. In the $DiSMO$, the best solution of the search region is considered as the lead solution and under the gravity drive of the lead solution, all the rest of the solutions are disrupted. Through this characteristic, the exploration proficiency of the SMO algorithm is improved. The proffered scheme is estimated through substantial experimental analysis and evaluated that for solving the continuous optimization problems, $DiSMO$ may turn into a decent decision. In future, the $DiSMO$ might be connected to understand some complex continuous real world optimization problems as well as its efficiency may also be evaluated for discrete optimization problems.

REFERENCES

[1] M Montaz Ali, Charoenchai Khompatraporn, and Zelda B Zabinsky. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization*, 31(4):635–672, 2005.

[2] 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.

[3] N Gouthamkumar, Veena Sharma, and R Naresh. Disruption based gravitational search algorithm for short term hydrothermal scheduling. *Expert Systems with Applications*, 42(20):7000–7011, 2015.

[4] Kavita Gupta, Kusum Deep, and Jagdish Chand Bansal. Improving the local search ability of spider monkey optimization algorithm using quadratic approximation for unconstrained optimization. *Computational Intelligence*, 2016.

[5] James Rauff et al. Nature-inspired optimization algorithms. *Mathematics and Computer Education*, 49(3):208, 2015.

[6] S Sarafrazi, H Nezamabadi-Pour, and S Saryazdi. Disruption: a new operator in gravitational search algorithm. *Scientia Iranica*, 18(3):539–548, 2011.

[7] Ajay Sharma, Harish Sharma, Annapurna Bhargava, and Nirmala Sharma. Optimal design of pida controller for induction motor using spider monkey optimization algorithm. *International Journal of Metaheuristics*, 5(3-4):278–290, 2016.

[8] Ajay Sharma, Harish Sharma, Annapurna Bhargava, and Nirmala Sharma. Power law-based local search in spider monkey optimisation for lower order system modelling. *International Journal of Systems Science*, pages 1–11, 2016.

[9] Ajay Sharma, Harish Sharma, Annapurna Bhargava, Nirmala Sharma, and Jagdish Chand Bansal. Optimal placement and sizing of capacitor using limaçon inspired spider monkey optimization algorithm. *Memetic Computing*, pages 1–21, 2016.

[10] Ajay Sharma, Harish Sharma, Annapurna Bhargava, Nirmala Sharma, and Jagdish Chand Bansal. Optimal power flow analysis using lévy flight spider monkey optimisation algorithm. *International Journal of Artificial Intelligence and Soft Computing*, 5(4):320–352, 2016.

[11] Nirmala Sharma, Harish Sharma, Ajay Sharma, and Jagdish Chand Bansal. Modified artificial bee colony algorithm based on disruption operator. In *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, pages 889–900. Springer, 2016.

[12] Urvinder Singh, Rohit Salgotra, and Munish Rattan. A novel binary spider monkey optimization algorithm for thinning of concentric circular antenna arrays. *IETE Journal of Research*, pages 1–9, 2016.

[13] Ponnuthurai N Suganthan, Nikolaus Hansen, Jing J Liang, Kalyanmoy Deb, Ying-Ping Chen, Anne Auger, and Santosh Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL report*, 2005005:2005, 2005.