Applied Mechanics and Materials
ISSN: 1662-7482, Vols. 380-384, pp 1216-1220
doi:10.4028/www.scientific.net/AMM.380-384.1216
© 2013 Trans Tech Publications, Switzerland
Online: 2013-08-30

A Novel Disruption Operator in Particle Swarm Optimization

Guiyan Ding, Hao Liu, Xiqin He

School of Science, University of Science and Technology Liaoning, China, 114051

Email: dinggy2006@sina.com, liuhustl@sina.cn

Keywords: Swarm intelligence; Particle swarm optimization; Disruption operator.

Abstract. Particle Swarm Optimization (PSO) has attracted many researchers attention to solve variant benchmark and real-world optimization problems because of its simplicity, effective performance and fast convergence. However, it suffers from premature convergence because of quickly losing diversity. To enhance its performance, this paper proposes a novel "disruption" strategy, originating from astrophysics, to shift the abilities between exploration and exploitation. The proposed Disruption PSO (DPSO) has been evaluated on a set of nonlinear benchmark functions and compared with other improved PSO. Comparison results confirm high performance of DPSO in solving various nonlinear functions.

Introduction

Numerical and combinational optimization problems arise in almost every field of science, engineering and business. Lots of these problems are NP-hard. PSO [1] is widely used to solve these problems because of its simplicity, fast convergence and high performance.

Researchers have proposed various modified versions of PSO to improve its performance, however, there still are premature or lower convergence rate problems. How to accelerate the convergence speed and how to avoid the local optimal solution are two important issues in the PSO research. Generally speaking, those issues could be classified the following three approaches [2]:

(1) Control of algorithm parameters: the inertia weight and the acceleration coefficients[3,4].

(2) Hybrid PSO, which combine PSO with other auxiliary search operators[5,6].

(3) Improvement of the topological structure[7,8].

Gravitational Search Algorithm (GSA) is a novel population algorithm based on the law of gravity firstly proposed by E. Rashedi, H. Nezamabadi-pour and S. Saryazdi [9]. To improve the exploration and exploitation ability, in [10], a disruption operator is introduced in GSA. The authors of this paper propoded an improved version in [11] to enhance its performance.

In this paper, we introduce an improved disruption operator into PSO with time-varying $V_{\max 1}$, and applied it on 10 nonlinear benchmark functions to confirm its high performance by comparing with other modified PSO.

Standard PSO (SPSO)

The PSO is inspired by the behavior of bird flying or fish schooling, it is firstly introduced by Kennedy and Eberhart in 1995 [1] as a new heuristic algorithm. In the PSO, a swam consists of a set of particles; and each particle represents a potential solution of an optimization problem. Considering the ith particle in the swarm with N particles in an n-dim space, its position and velocity at iteration t are denoted by $X_i(t)=(x_1(t),\cdots,x_n(t))$ and $V_i(t)=(v_1(t),\cdots,v_n(t))$. Then, the new velocity and position on the d-dimension of this particle at iteration $t+1$ will be calculated by using the following equations:

$$\begin{aligned} v_i^d(t+1)=w\cdot v_i^d(t)+r_1\cdot c_1\cdot\left(Pbest_i^d(t)-x_i^d(t)\right) \\ +r_2\cdot c_2\cdot\left(Gbest^d(t)-x_i^d(t)\right) \end{aligned} \tag{1}$$

$$x_i^d(t+1)=x_i^d(t)+v_i^d(t+1) \tag{2}$$

Where w is the inertial weight to balance the global and local search abilities of particles in the search space; r_1 and r_2 are two uniformly distributed random numbers in the interval [0,1]; $Pbest_i(t)=\left(Pbest_{i,1}(t),\cdots,Pbest_{i,n}(t)\right)$, called the personal best solution (position), represents the best solution found by the *i*th particle itself until iteration *t*; $Gbest(t) = (Gbest_1(t), \cdots, Gbest_n(t))$, called the global best solution, represents the global best solution found by all particles until iteration *t*; acceleration coefficients c_1 and c_2 are nonnegative constants which control the influence of the cognitive and social components on the search process.

Disruption PSO (DPSO)

To improve the exploration and exploitation abilities of SPSO, a novel operator called "Disruption", originating from astrophysics, will be introduced in SPSO. This paper, we will proposed an improved disruption operator.

To simulate the disruption phenomenon: " When a swarm of gravitational bound particles having a total mass, m, approaches too close to a massive object, M, the swarm tends to be torn apart. The same thing can happen to a solid body held together by gravitational forces when it approaches a much more massive object" [12]. it is assumed that $Gbest(t)$ is the star of all particles, and the other solutions can potentially disrupt and scatter in the search space.

A. Disruption Condition

Whether all particles except $Gbest(t)$ satisfy the disruption condition Eq. (3) or not. $\frac{R_{ij}}{R_{ibest}} < C$ (3)

where R_{ij} and R_{ibest} are Euclidean distances between particles i and j and between particle i and the star, respectively. It is noted that particle j is the neighbor of particle i, whose fitness is the just better than partcle i in the search space, i.e. after sorting the fitness values, $j = i - 1$.

B. Disruption Operator

To make full use of R_{ij}, we let disruption operator as follows:

$$D = \begin{cases} R_{ij} \times U\left(-\frac{R_{ij}}{2}, \frac{R_{ij}}{2}\right) & if\ R_{ibest} \ge 1 \\ R_{ij} + U\left(-\frac{1}{2}, \frac{1}{2}\right) & otherwise. \end{cases} \tag{4}$$

In this equation, $U\left(-\frac{R_{ij}}{2}, \frac{R_{ij}}{2}\right)$ return a uniformly distributed random number in the interval $\left[-\frac{R_{ij}}{2}, \frac{R_{ij}}{2}\right]$. The disruption operator explores initially and as time passes, it switches to the exploiting condition. We denote the PSO equipped with disruption operator by DPSO.

C. Disruption Strategy

To enhance performance, we move those particles, which satisfy the disrupton condition, by using the following disruption strategy :

$$X_i(t) = \frac{t}{T} \cdot X_i(t) + \left(1 - \frac{t}{T}\right) \cdot X_i(t) \cdot D \tag{5}$$

Table 1. The Pseudo-code of the DPSO

DPSO Algorithm
Initialize an *n*-dimensional swarm, *N*
repeat
for each particle $i = 1, 2, \cdots, N$
if X_i exceeds the allowable range
then limits X_i to the boundary value;
Evaluate fitness value;
Update $Pbest_i$;
Update $Gbest$;
Update the velocity using Eq. (1);
Time-varying V_{max1} : $V_{max1} = e^{-\frac{t}{T}} \cdot V_{max}$;
if V_i exceeds the allowable range $[-V_{max1}, V_{max1}]$
then limits V_i to the boundary value;
Update the position using Eq. (2);
Disrupt particles using Eq. (3), (4), (5);
End
Until stopping condition is met;

Table 2. The 6 unimodal functions used in experimental studies, where n is the dimension of the functions, f is the minimum values of the functions, and $X \subseteq R^n$ is the search space.

Test Function	*n*	*X*	*f*
$F_1(x)=\sum_{i=1}^{n} x_i^2$	20	$[-5.12,5.12]^n$	0
$F_2(x)=\sum_{i=1}^{n} i\cdot x_i^2$	20	$[-5.12,5.12]^n$	0
$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	20	$[-65.536,65.53$	0
$F_4(x)=\sum_{i=1}^{n}\lvert x_i\rvert^{i+1}$	20	$[-1,1]^n$	0
$F_5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(1-x_i\right)^2\right]$	30	$[-30,30]^n$	0
$F_6(x)=\sum_{i=1}^{n} i\cdot x_i^4+random[0,1)$	30	$[-1.28,1.28]^n$	0

Table 3 The 4 multimodal functions used in experimental studies, where n is the dimension of the functions, f is the minimum values of the functions, and $X \subseteq R^n$ is the search space.

Test Function	*n*	*X*	*f*
$F_7(x)=\sum_{i=1}^{n}-x_i\sin\left(-\sqrt{\lvert x_i\rvert}\right)$	30	$[-500,500]^n$	-12569.5
$F_8(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	30	$[-5.12,5.12]^n$	0
$F_9(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	$[-600,600]^n$	0
$F_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	30	$[-32,32]^n$	0

Table 4 The specific parameter settings

Test Function	*Number of Generations*	*Population size N*	$c_1=c_2$	w
F_1-F_4	1000	10	1.49618	0.72984
F_5	10000	50	1.49618	0.72984
F_6	3000	50	1.49618	0.72984
F_7-F_8	5000	50	1.49618	0.72984
F_9-F_{10}	1000	50	1.49618	0.72984

Benchmark Functions and Parameter Settings

To evaluate the performance of DPSO, it is applied to 10 well-known benchmark functions used in [6]. Table 2 and 3 list the 10 test functions. They are high-dimensional problems and divided into two classes: unimodal and multimodal problems. In which functions, F_1 to F_6 in table 1 are unimodal functions, and functions F_7 to F_{10} in Table 2 are multimodal functions. All the functions used in this paper are minimization problems.

The selection of the parameters w, c_1, c_2 of Eq.(1) is very important. It can greatly influence the performance of PSO algorithms and its variations, We set w, c_1 and c_2 in Table 4 like [6]. However, there are a few differences for different problems. In this paper, $V_{\max}$ is all related PSO algorithms is set to 2.0 and the stopping criteria is set to the maximal generations T.

In order to compare the different algorithms, the same settings have been used in HPSO [6], the same maximal generations and the same population size were used. In HPSO, $W_{\max}$ and N are set to 1 and 20 respectively. In DPSO, the threshold $C=C_0\left(1-\frac{t}{T}\right)$ with $C_0=100$.

Experimental results and Discussion

To verify the high performance of DPSO, we Compare it with SPSO and the best HPSO in [6].

Table 5 shows the results. All results are averaged over 50 runs, where "***Ave***" indicates the average of best fitness values found in the last generation, and "***Std Dev***" stands for the standard deviation. All data in terms of HPSO directly come from [6].

As Table 5 illustrates, statistically speaking, for the average of best fitness values of 50 runs on 10 test functions, DPSO enhanced the performance obviously than SPSO and HPSO on all test functions. Especially, you can see clearly that DPSO obtains the global optimum on F_4, F_7, F_8 and F_9 . In terms of "***Std Dev***", we also can see clearly that, DPSO shows high stability than SPSO and HPSO. DPSO can obtain 0 "Std Dev" on F_1 to F_4, F_8 and F_9. Therefore, DPSO is of the best performance and convergence. Both for the unimodal functions (F_1 to F_6) and multimodal functions (F_7 to F_{10}), DPSO obtains a promising performance.

In terms of convergence charcteristics, Fig. 1 concretely shows the comparison on selected functions between DPSO and SPSO. Roughly speaking, DPSO has extremely enhanced the convergence speed. For the simple unimodal functions, DPSO and SPSO performed equally well at the beginning (about 100 generations) , because the particles at that time are of satisfactory diversity so that both methods could improve well. Once the particles in the population are close to the best particle, the convergence of SPSO becomes slower because the diversity of the population will decrease. With the help of Disruption Operator on those partcles, which satisfy the disruption condition, DPSO could move these particles away from the best particle, so the diversity will be enhanced, therefore, the fast speed could remain through the whole evolution process. For the difficult multimodal functions, Disruption Operator also could move those particles, which satisfy the disruption condition, away from the local minimum. At the same time, the decreasing exponentially maximal velocity, reduces the speed rate to local minimum, and enhances the probability of disruption. To sum up, DPSO could successfully find better solutions while enhancing convergence speed.

TABLE 5. MINIMIZATION RESULTS OF BENCHMARK FUNCTIONS

F	SPSO		HPSO[6]		DPSO	
	Ave	***Std Dev***	***Ave***	***Std Dev***	***Ave***	***Std Dev***
F_1	3.61e-8	9.87e-8	1.79e-7	3.51e-7	**2.3e-217**	**0**
F_2	8.49e-4	0.0027	6.38e-7	1.98e-6	**4.7e-215**	**0**
F_3	2.6521	4.6355	0.398	0.3082	**5.8e-190**	**0**
F_4	1.59e-28	4.98e-28	2.53e-19	9.38e-19	**0**	**0**
F_5	2.2156	2.6799	1.419	1.4256	**4.86e-5**	**4.25e-5**
F_6	0.0060	0.0023	4.37e-3	1.51e-3	**7.3566e-6**	**5.38e-6**
F_7	-5.72e+3	1.62e+3	-12558.9	6.2373	**-1.2569e+4**	**8.05e-5**
F_8	52.8322	8.2374	31.8005	9.1618	**0**	**0**
F_9	0.3288	0.3599	3.66e-2	3.19e-2	**0**	**0**
F_{10}	0.1155	0.3653	8.86e-6	8.58e-2	**1.59e-15**	**1.49e-15**

Conclusion

In this paper, we introduce a disruption operator in standard PSO, which is called DPSO, to help SPSO to shift the exploration and exploitation abilities and avoid local optima. We also proposed time-varying $V_{\max 1}$, which decreasing exponentially with time t, to control the flying velocity. By applying a disruption operator on those particles, which satisfy the disruption condition in each generation, DPSO could find better solutions than SPSO and HPSO.

DPSO is evaluated on 10 well-known nonlinear functions. The results have proved that DPSO could have faster convergence and better global search ability both on unimodal functions and multimodal functions compared to SPSO and HPSO. DPSO also performs better stability than SPSO and HPSO.

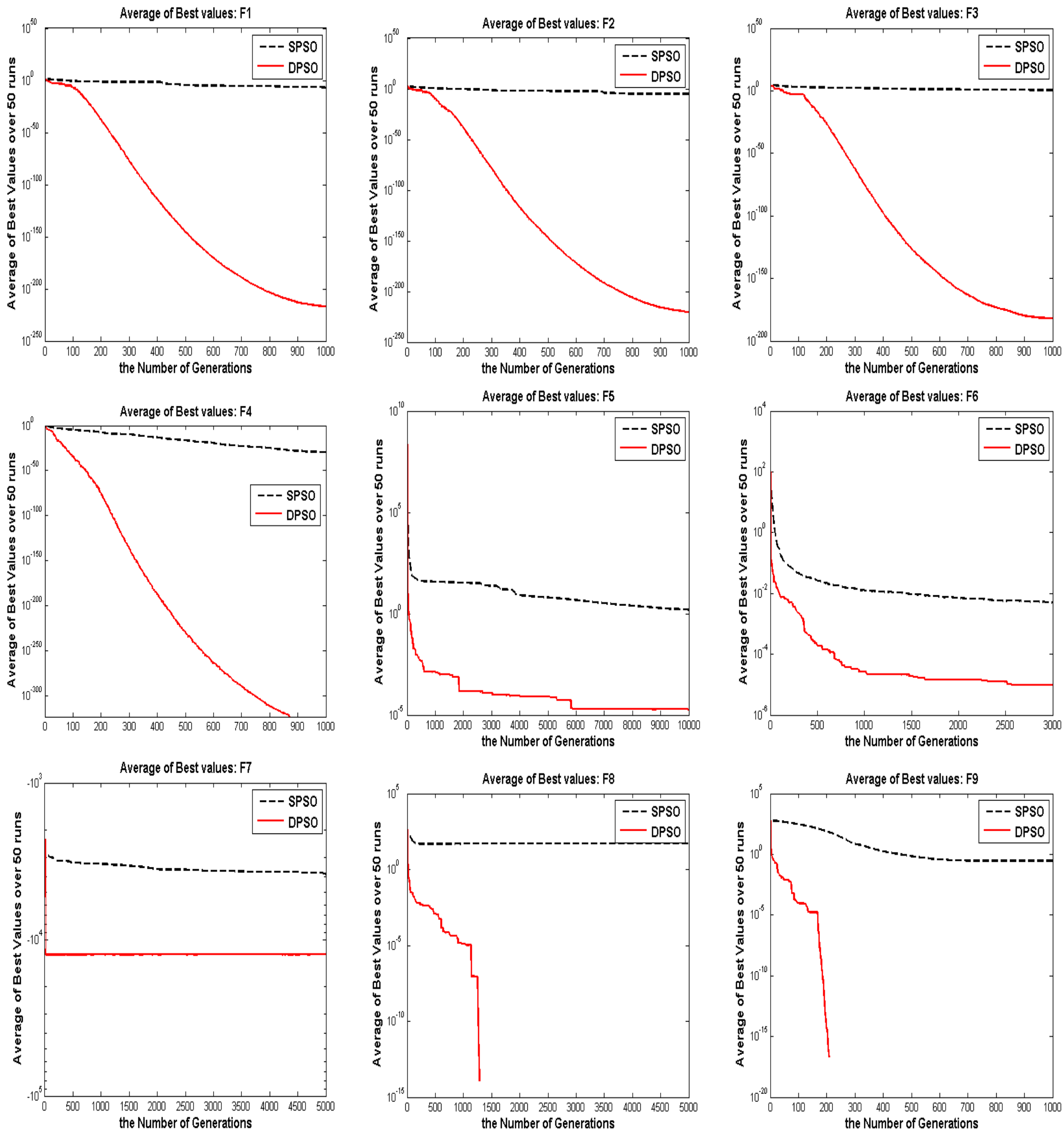

Figure 1. Comparison between SPSO and DPSO on F_1 to F_9.

References

[1]J. Kennedy and R. Eberhart, Particle Swarm Optimization, IEEE International Conference on Neural Networks, Perth, Australian. pp. 1942-1948, 1995.

[2]Z.H. Zhan, J.Z. Zhan, Y. Li, H.S.H. Chung, Adaptive particle swarm optimization, IEEE Transactions on Systems, Man and Cybernetis Part B: Cybernetics, vol. 39, no. 6, pp. 1362-1381, 2009.

[3]Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proc. IEEE World Congr. Comput. Intell., pp. 69-73, 1998.

[4]A. Ratnaweera, S. Halgamuge, H. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 240-255, 2004.

[5]Y.P. Chen, W.C. Peng, M.C. Jian, Particle swarm optimization with recombination and dynamic linkage discovery, IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetis, vol. 37, no. 6, pp. 1460-1470, 2007.

[6] H. Wang, Y. Liu, C.H. Li, S.Y. Zeng, A hybrid particle swarm algorithm with cauchy mutation, IEEE Swarm Intelligence Symposimu 2007 (SIS 2007), Honolulu, Huawaii, USA, 2006.

Vehicle, Mechatronics and Information Technologies
10.4028/www.scientific.net/AMM.380-384

A Novel Disruption Operator in Particle Swarm Optimization
10.4028/www.scientific.net/AMM.380-384.1216