Cat Swarm Optimization with a Vibrational Mutation Strategy

Yan Zhang and Yide Ma

Abstract—**Cat swarm optimization (CSO), a relatively new swarm intelligence algorithm, exhibits better performance on optimization problems than particle swarm optimization (PSO) and weighted-PSO. This paper presents a variation on the standard CSO algorithm called a vibrational mutation cat swarm optimization, or VMCSO in order to efficiently increase diversity of the swarm in the global searches. Comparing the new algorithm with CSO and several CSO main variants demonstrates the superiority of the VMCSO for the benchmark functions.**

Index Terms—**Cat Swarm Optimization, Vibrational mutation, Diversity, Swarm intelligence.**

I. Introduction

A new swarm intelligence algorithm called cat swarm optimization which imitates the natural behavior of cats was proposed by Chu and Tsai [1]. It is one of the swarm intelligence algorithms which is the collective behavior of decentralized, self-organized system, natural or artificial [2]. The number of its successful applications is growing in clustering [3], networks [4], [5], solving multi-objective problems [6], image edge enhancement [7].

The CSO method has two modes, i.e., the tracing mode and the seeking mode, for simulating the behaviors of cats to move the individuals in the solution space. A mixture ratio MR is used to set the cats into tracing mode and seeking mode. If MR is equal to 0, all the cats will move into the seeking mode; If MR is equal to 1, all the cats will move into the tracing mode. In most cases, MR is set a small value so that most cats are in the seeking mode. In the seeking mode, the cats spend most of time resting but being alert looking around their environment for the next move.

The seeking mode represents the local searching and the mutation operator in this mode provides the necessary local variety within the cats. In the tracing mode, the cats exhibit similar behaviors with the particle swarm Optimization (PSO) [8]. The particles in PSO update their positions according to the value of the global best and the local best but the cats in tracing mode update their positions only according to the value of the global best because the local mutation is utilized in the seeking mode.

The tracing mode represents the global searching and it provides attractive features such as the higher convergence speed. Most of the modifications of CSO have been proposed so far focusing on the tracing mode to overcome the issues related to convergence speed, prematurely converged solutions, deficient accuracy and so on [9], [10].

II. Definition and Mathematical Analysis of Basic CSO Algorithms Introduction

We apply the cats into the CSO to solve problem, the number of the cats used in the iteration is the swarm size. Each cat of the swarm has a current position composed of M dimensions, a current velocity, a fitness value that represents the accommodation of it to the fitness function, and a flag called MR(mixture ratio) to identify whether the cat is in seeking mode or tracing mode.

The frame of CSO algorithm [1] can be described as the following steps:

Step 1: Create N cats as the swarm.

Step 2: Randomly assign the position in the M-dimensional solution space to the cats and randomly select velocity of each cat. The position and the velocity are in-range of the maximum value. Haphazardly pick numbers of cats according to MR.

Step 3: Evaluate the cats according to the fitness function and keep position of the best cat into memory.

Step 4: Apply the cat into the seeking mode process when its MR is 1 and apply the cat into the tracing mode process when its MR is 0.

Step 5: Randomly re-pick number of cats and set them into seeking mode and the others into tracing mode.

Step 6: Check the stop condition, if met, stop the program, and otherwise repeat the Step 3 to Step 5.

A. Seeking Mode

The seeking mode corresponds to a local search process of the optimization problem. The seeking mode has four essential factors: seeking memory pool (SMP) which is the copies of a cat produced in the seeking mode, seeking range of the selected dimension (SRD) which is the mutative ration for the selected dimensions, counts of dimension to change (CDC) and self position consideration (SPC) [11] .The action of the seeking mode is described as follows:

Step 1: Create SMP copies of $\text{cat}_{k.}$

Step 2: According to CDC, apply the mutation operator to the T copies. Randomly minus or plus SRD percents the present values, replace the old values.

Step 3: Evaluate the fitness of the mutated copies, if all FS are not exactly equal, use equation (1) to calculate the selecting probability of each candidate, otherwise set all be 1.

Manuscript received April 7, 2014; revised June 22, 2014. This work was jointly supported by National Natural Science Foundation of China (No.61175012& 61201421), Natural Science Foundation of Gansu Province (No. 1208RJZA265), Specialized Research Fund for the Doctoral Program of Higher Education of China (No.20110211110026), the Fundamental Research Funds for the Central Universities of China (No.lzujbky-2013-k06)

Yan Zhang and Yide Ma are with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China (corresponding author : Yide Ma; e-mail: zhangaoliao@gmail.com, yidema@gmail.com).

DOI: 10.7763/IJMLC.2014.V4.464

FS is for the fitness value.

$$P_i = \frac{|FS_i - FS_b|}{FS_{\max} - FS_{\min}}, \quad \text{where} \quad 0 < i < j \tag{1}$$

Step 4: Pick the point to move to from the candidate points randomly, and replace the position of cat$_k$.

Submit your manuscript electronically for review.

B. Tracing Mode

The tracing mode corresponds to a global search process of the optimization problem. The action of the tracing mode is described as follows:

Step 1: Update the velocities of each dimension of a cat using the equations (2).

Step 2: Check if the velocities are in the range of maximum and minimum velocities, if it is over-range, set it be equal to the limit.

Step 3: Update the positions using the equations (3).

$$v_{k,d,t} = v_{k,d,t-1} + r_1 \times c_1 \times \left(x_{best,d,t-1} - x_{k,d,t-1}\right), \tag{2}$$

where $d = 1, 2, \cdots, M$

$$x_{k,d,t} = x_{k,d,t-1} + v_{k,d,t} \tag{3}$$

where $x_{best,d,t}$ is the position of the cat$_k$ who has the best fitness value at iteration t, $x_{k,d,t}$ is the position of the cat$_k$ at iteration t, every cat has its own velocity and position composed of M dimensions. c_1 is a constant and r_1 is a random value in the range of [0, 1].

Let us analyze the equation (1) and (2) in an analytical way. For the sake of the simple analysis, assume that $c_1 \cdot r_1$ is constant that is equal to φ_1 and $x_{best,d,t-1}$ is also constant, such as x_b. Then, (2) becomes

$$v_{k,d,t} = v_{k,d,t-1} + \varphi_1 \times \left(x_b - x_{k,d,t-1}\right) \tag{4}$$

By getting $v_{k,d,t}$ and $v_{k,d,t-1}$ using (3), and substituting them into (4), the following position equation is obtained:

$$x_{k,d,t} - x_{k,d,t-1} = x_{k,d,t-1} - x_{k,d,t-2} + \varphi_1 \times \left(x_b - x_{k,d,t-1}\right) \tag{5}$$

Eq. (5) can be recast as

$$x_{k,d,t} - (2-\varphi_1)x_{k,d,t-1} + x_{k,d,t-2} = \varphi_1 x_b \tag{6}$$

Eq. (6) is a linear nonhomogeneous second-order difference equation. The general solution of (6) can be derived by getting complementary and particular solutions. It has a corresponding homogeneous equation:

$$x_{k,d,t} - (2-\varphi_1)x_{k,d,t-1} + x_{k,d,t-2} = 0 \tag{7}$$

The general solution can be expressed in the form

$$x_{k,d,t} = X_{k,d,t} + X^*_{k,d,t} \tag{8}$$

where $X_{k,d,t}$ is the complementary solution and $X^*_{k,d,t}$ is the particular solution. The auxiliary equation is:

$$\lambda^2 - (2-\varphi_1)\lambda + 1 = 0 \tag{9}$$

with roots:

$$\lambda = \frac{2-\varphi_1 \pm \sqrt{(2-\varphi_1)^2 - 4}}{2} = \frac{2-\varphi_1 \pm \sqrt{\varphi_1^2 - 4\varphi_1}}{2} \tag{10}$$

This result may expose three different cases depending on the value of $\varphi_1^2 - 4\varphi_1$. Generally, the value of c_1 is 2, and $2 > \varphi_1 > 0$, so $\varphi_1^2 - 4\varphi_1 \le 0$, the roots is:

$$\lambda = \frac{2-\varphi_1 \pm i\sqrt{4\varphi_1 - \varphi_1^2}}{2} = \alpha \pm i\beta \tag{11}$$

The solution of the complementary equation is:

$$X_{k,d,t} = r^t \left(A_1 \cos\theta t + A_2 \sin\theta t\right) = A_1 \cos\theta t + A_2 \sin\theta t$$

where

$$r = \sqrt{\alpha^2 + \beta^2} = 1 \text{ , } \tan\theta = \beta/\alpha . \tag{12}$$

For a particular solution we try $X^*_{k,d,t} = K$. Then $X^*_{k,d,t-1} = K$ and $X^*_{k,d,t-2} = K$. Substituting into the difference equation (5), we have

$$K - (2-\varphi_1)K + K = \varphi_1 x_b \tag{13}$$

So $\varphi_1 K = \varphi_1 x_b$, $K = x_b$, thus, the particular solution is $X^*_{k,d,t} = x_b$.

And the general solution is

$$x_{k,d,t} = A_1 \cos\theta t + A_2 \sin\theta t + x_b \tag{14}$$

Some assumptions are made such that there is one dimension, $x^{(0)} = x_0$, $v^{(0)} = v_0$. Substituting them into (14), then

$$x_0 = A_1 + x_b \text{ , } A_1 = x_0 - x_b$$
$$v^{(1)} = v_0 + \varphi_1\left(x_b - x_0\right)$$
$$x^{(1)} = x_0 + v^{(1)} = x_0 + v_0 + \varphi_1\left(x_b - x_0\right) =$$
$$\left(x_0 - x_b\right)\cos\theta + A_2 \sin\theta + x_b$$

where $\cos\theta = (2-\varphi_1)/2$, $\sin\theta = \sqrt{4\varphi_1 - \varphi_1^2}\,/\,2$

Then $A_2 = \left(2v_0 + x_b\varphi_1 - x_0\varphi_1\right)/\sqrt{4\varphi_1 - \varphi_1^2}$.

From (14), we can conclude that the path of cat$_k$ can be considered as an oscillation phenomenon around the point x_b. They surf on an underlying foundation of sine waves, so the drawback of the tracing mode is due to the lack of diversity in the global searching. This similar phenomenon in PSO was discussed in [12].

If an optimization algorithm is no longer able to explore sections of the search space other than the examined area, it converges prematurely to a local optimum because another region may exists a solution superior to the currently discovered one. So the diversity is an essential factor for a good search. At this point, we could design a new mutation operator called a global vibrational mutation strategy for the positions updating equation (3) in the tracing mode. The details of the algorithm will be described in subsequent sections.

C. VMCSO

Let us proceed with (14) and add some changes as follows:

$$x_{k,d,t} = \begin{cases} x_{k,d,t} + A_n\delta & \delta = 1\, if\ t > nT, m = 1, \\ x_{k,d,t} - A_n\delta & \delta = 0\, if\ t > nT, m = 0, \\ x_{k,d,t} & t < nT, n = 1,2,3\cdots \end{cases} \tag{15}$$

Here, δ is the delta function, T is the period of the sine waves and A_n is the amplitude factor that defined by user, it is computed during the generations and defined as the first amplitude of the current period. m is the parity flag, when the number of the period is odd, it is set 0, otherwise it is set 1. Fig. 1 shows the resulting graphs for (14) and (15) and T is 30.

Ultimately equation (15) replaces equation (3) in the tracing mode.

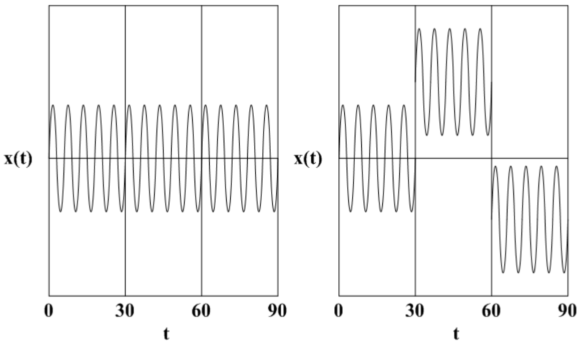

Fig. 1. Trajectory of a cat in CSO with (14) (left) and (15) (right).

III. Experimental Verification and Comparisons

A. Functions Tested and CSOs Compared

Six benchmark functions including unimodal and multimodal which listed in Table I are used in the experimental tests. These benchmark functions are widely used in benchmarking global optimization algorithms [13], [14].

TABLE I: Six Test Functions Used in the Comparisons

Function	Formula	Opt.position	Opt.value	Trait	Search Range
Sphere	$f_1(x) = \sum_{j=1}^{n} x_j^2$	$(0,0)$	0	U	$[-5.12,5.12]^D$
Rosenbrock	$f_2(x) = \sum_{j=1}^{n}\left[100\left(x_{j+1} - x_j^2\right)^2 + \left(1 - x_j\right)^2\right]$	$(1,1,\ldots,1)$	0	U	$[-10,10]^D$
Rastrigrin's	$f_3(x) = \sum_{j=1}^{n}\left[x_j^2 - 10\cos\left(2\pi x_j\right) + 10\right]$	$(0,0,\cdots,0)$	0	M	$[-5.12,5.12]^D$
Schwefel's	$f_4(x) = 418.9829n + \sum_{i=1}^{n}\left[-x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)\right]$	$(0,0,\cdots,0)$	0	M	$[-500,500]^D$
Griewangk's	$f_5(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$(0,0,\cdots,0)$	0	M	$[-600,600]^D$
Ackley's	$f_6(x) = -20\cdot\exp\left(-0.2\cdot\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right) + 20 + e$	$(0,0,\cdots,0)$	0	M	$[-32.768,32.768]^D$

U: unimodal. M: multimodal. D: dimensions.

In literature, some algorithms were proposed to improve the performance of the CSO since 2006. The standard CSO and two CSOs are selected as comparative algorithms to make a comparison, they are parallel cat swarm optimization (PCSO) [9] and average-inertia weighted CSO (AICSO) [10].

In PCSO, a parallel structure in the tracing mode that changed the cats' behavior is used. The individuals are separated into two or four sub-populations. They move forward to the local best solution of its own group in general instead that move forward to the global best solution directly. If the programmed iteration achieved, the local best solutions at present are stopped and a sub-population is randomly picked to replace the worst individual in the selected sub-population, the information is exchanged once per ECH iteration [9].

In AICSO, a parameter called inertia weight in the velocity update equation is used and a new form of the position update equation is proposed. The cat's velocity and its position are updated by the following equations:

$$v_{k,d,t+1} = wv_{k,d,t} + r_1 \times c_1 \times \left(x_{best,d,t} - x_{k,d,t}\right) \tag{16}$$

$$x_{t+1} = \frac{\left(x_t + x_{t-1}\right)}{2} + \frac{\left(v_t + v_{t-1}\right)}{2} \tag{17}$$

The inertia weight w is decreased from 0.9 to 0.4 during the iterations [5].

B. Performance Evaluation

Table II gives the parameter settings for all the algorithms, for a fair comparison, most parameters are set same according to the settings in references. The test procedure includes two bundles. In the first bundle, the algorithms are tested using the same population size of 50 with different dimensions as D is equal to 10, 20 and 30. The max iteration is fixed as 10000. The algorithm sensitivity to different dimensions is tested. In the second bundle, the algorithms are tested using the different population size such as 10, 20 and 30 with fixed dimensions as D is 30. The max iteration is fixed as 10000. The algorithm sensitivity to different population size is tested.

TABLE II: PARAMETERS SETTING IN THE COMPARISONS

Algorithm	num	SMP	CDC	SRD	c_1	MR	ECH	w	T
CSO	50	5	1	0.2	2	20%	×	×	×
PCSO(2 group)	50	5	1	0.2	2	20%	20	×	×
PCSO(4 group)	50	5	1	0.2	2	20%	20	×	×
AICSO	50	5	1	0.2	2.05	20%	×	0.9 to 0.4	×
VMCSO	50	5	1	0.2	2	20%	×	×	50

In Table III, the numerical results for six test functions of the first bundle are shown. A total of 100 runs for each experiment are conducted. Mean fitness and standard deviation of the final solutions are used to represent the quality of the results. The best results among all the algorithms are shown in bold. For all the functions, the results show that VMCSO outperforms the classical CSO. For *Rastrigrin* , *Schwefel's* , *Griewangk's* and *Ackley's* functions, VMCSO outperforms other CSO variants, its mean best values and standard deviation of the final solutions are less. It can be seen that VMCSO is considerably superior to other CSO variants for *Sphere* function, but PCSO(4 group) gets better results, however, VMCSO's result is also satisfying. As the problem becomes more complex and become multimodal in high dimension, the performance of VMCSO is much better in CSOs because the cats could search for a new region to find a better solution and escape from the local optima.

TABLE III: RESULTS FOR 6 TEST FUNCTIONS IN 10, 20, 30 DIMENSIONS

Method		CSO	PCSO(2 group)	PCSO(4 group)	AICSO	VMCSO
f	D	Mean(std.dev)	Mean(std.dev)	Mean(std.dev)	Mean(std.dev)	Mean(std.dev)
$f_1(x)$	10	7.68E-83±(1.52E-82)	1.45E-84±(1.78E-84)	7.38E-98±(1.48E-97)	7.48E-84±(1.31E-83)	**4.81E-101±(6.30E-101)**
	20	1.64E-30±(6.97E-30)	2.99E-79±(5.97E-79)	**3.52E-107±(4.44E-107)**	1.28E-34±(1.92E-34)	6.29E-85±(1.26E-84)
	30	5.35E-23±(1.09E-23)	2.90E-83±(5.79E-83)	**2.58E-89±(5.15E-89)**	8.38E-25±(1.61E-24)	8.13E-82±(1.51E-81)
$f_2(x)$	10	2.91E+00±(3.17E+00)	4.88E+00±(2.15E+00)	**2.36E+00±(1.08E+00)**	4.42E+00±(4.32E+00)	3.76E+00±(3.07E+00)
	20	5.88E+00±(3.87E+00)	9.63E+00±(9.57E+00)	4.67E+00±(3.35E+00)	1.49E+01±(1.01E+01)	**4.13E+00±(6.57E+00)**
	30	2.91E+01±(9.86E+00)	1.49E+01±(1.63E+01)	1.43E+01±(1.32E+01)	1.64E+01±(1.19E+01)	**8.05E+00±(1.12E+01)**
$f_3(x)$	10	**0.00E+00±(0.00E+00)**	1.33E-10±(8.56E-11)	3.70E-08±(5.77E-08)	**0.00E+00±(0.00E+00)**	**0.00E+00±(0.00E+00)**
	20	5.71E+00±(7.83E+00)	3.88E-01±(3.64E-01)	4.38E+00±(7.85E+00)	4.20E+00±(8.41E+00)	**2.48E-01±(3.98E+00)**
	30	8.59E+00±(1.15E+01)	1.55E+00±(1.54E+00)	3.57E+00±(2.73E+00)	6.71E+00±(1.32E+01)	**7.71E-01±(5.42E+00)**
$f_4(x)$	10	3.14E-01±(3.68E-01)	8.56E-01±(5.21E-01)	5.70E-01±(2.35E-01)	2.16E-01±(4.60E-01)	**1.55E-01±(1.83E-01)**
	20	2.93E+00±(3.74E+00)	7.30E-01±(4.33E-01)	2.35E+00±(1.19E+00)	3.48E-01±(2.16E-01)	**2.51E-01±(2.30E-01)**
	30	2.65E+00±(1.56E+00)	3.09E+00±(2.99E+00)	7.16E+02±(1.42E+03)	3.06E+00±(2.93E+00)	**2.05E+00±(9.03E-01)**
$f_5(x)$	10	**1.00E+00±(0.00E+00)**	**1.00E+00±(0.00E+00)**	**1.00E+00±(0.00E+00)**	**1.00E+00±(0.00E+00)**	**1.00E+00±(0.00E+00)**
	20	1.00E+00±(3.92E-05)	1.00E+00±(3.29E-04)	1.00E+00±(3.29E-04)	1.00E+00±(2.47E-04)	**1.00E+00±(0.00E+00)**
	30	1.00E+00±(1.67E-04)	1.00E+00±(1.90E-03)	1.00E+00±(1.90E-03)	1.00E+00±(1.20E-03)	**1.00E+00±(0.00E+00)**
$f_6(x)$	10	1.05E-14±(3.48E-15)	1.77E-04±(2.90E-04)	8.60E-03±(6.60E-03)	3.37E-15±(1.42E-15)	**7.70E-16±(7.45E-15)**
	20	6.84E-01±(4.92E-01)	1.38E+00±(2.01E-01)	1.13E+00±(5.80E-01)	4.56E-01±(9.12E-01)	**7.01E-12±(6.63E-12)**
	30	1.73E+00±(9.12E-01)	1.27E+00±(9.74E-01)	1.65E+00±(8.69E-01)	3.18E-01±(6.14E-01)	**4.48E-10±(3.29E-10)**

TABLE IV: RESULTS FOR 6 TEST FUNCTIONS WITH 10, 20, 30 SWARM SIZE

Method		CSO	PCSO(2 group)	PCSO(4 group)	AICSO	VMCSO
f	P	Mean(std.dev)	Mean(std.dev)	Mean(std.dev)	Mean(std.dev)	Mean(std.dev)
$f_1(x)$	10	5.18E-50±(6.96E-50)	1.17E-59±(2.36E-59)	7.10E-70±(6.67E-70)	2.11E-20±(4.21E-20)	**3.98E-83±(7.96E-83)**
	20	3.46E-47±(7.69E-47)	3.71E-70±(7.27E-70)	3.18E-68±(6.35E-68)	1.53E-25±(2.94E-25)	**7.55E-80±(1.51E-79)**
	30	4.27E-45±(6.00E-45)	3.25E-69±(6.50E-69)	1.50E-67±(2.99E-67)	1.53E-24±(3.06E-24)	**1.36E-79±(2.73E-79)**
$f_2(x)$	10	4.65E+01±(3.22E+01)	3.68E+01±(7.51E+00)	3.84E+01±(1.28E+00)	4.94E+01±(3.44E+00)	**3.36E+01±(1.49E+00)**
	20	4.08E+01±(5.32E+01)	2.86E+01±(2.22E+01)	4.40E+01±(1.51E+01)	2.61E+01±(2.52E+00)	**1.39E+01±(1.31E+00)**
	30	3.46E+01±(1.41E+01)	2.51E+01±(3.22E+01)	3.19E+01±(1.70E+01)	1.56E+01±(1.09E+01)	**1.28E+01±(1.14E+01)**
$f_3(x)$	10	3.78E+01±(4.60E+01)	6.38E+00±(3.12E+00)	**5.38E+00±(4.19E+00)**	6.79E+01±(4.81E+00)	6.13E+01±(5.40E+01)
	20	3.37E+01±(2.33E+01)	3.04E+01±(2.05E+01)	7.03E+01±(6.65E+01)	5.26E+01±(1.94E+01)	**2.13E+01±(1.92E+01)**
	30	1.77E+01±(8.81E+00)	**2.70E+00±(1.51E+00)**	4.66E+00±(1.23E+00)	1.03E+01±(2.05E+01)	3.14E+01±(2.58E+01)
$f_4(x)$	10	8.95E+00±(1.78E+00)	6.31E+00±(2.61E+00)	4.56E+00±(7.01E+00)	7.68E+00±(5.68E+00)	**4.31E+00±(2.88E+00)**
	20	**1.23E+00±(2.45E+00)**	2.64E+00±(1.51E+00)	2.60E+00±(1.76E+00)	5.64E+00±(1.01E+00)	1.64E+00±(2.43E+00)
	30	8.83E+00±(1.81E+00)	1.12E+00±(1.42E+00)	1.09E+00±(1.45E+00)	3.12E+00±(7.49E+00)	**1.09E+00±(1.06E+00)**
$f_5(x)$	10	1.21E+00±(5.64E-04)	1.01E+00±(4.60E-06)	1.00E+00±(5.64E-08)	1.01E+00±(5.67E-04)	**1.01E+00±(6.60E-09)**
	20	1.11E+00±(3.62E-05)	1.00E+00±(3.79E-07)	1.00E+00±(3.62E-08)	1.00E+00±(3.62E-05)	**1.00E+00±(2.85E-09)**
	30	1.08E+00±(1.47E-06)	1.00E+00±(1.66E-07)	**1.00E+00±(1.47E-09)**	1.00E+00±(1.47E-06)	1.00E+00±(1.67E-09)
$f_6(x)$	10	3.45E+00±(1.23E+00)	1.76E+00±(1.18E+00)	3.33E+00±(1.53E-01)	2.56E+00±(1.58E+00)	**1.08E+00±(2.30E-01)**
	20	3.32E+00±(1.03E+00)	3.20E+00±(2.20E+00)	2.45E+00±(1.08E+00)	2.47E+00±(1.26E+00)	**1.42E+00±(1.35E-01)**
	30	3.08E+00±(4.63E+00)	3.11E+00±(6.73E+00)	2.15E+00±(1.66E+00)	1.45E+00±(1.88E+00)	**1.25E+00±(1.23E-01)**

The plots in Fig.2. shows the convergence progress of the global best values of the *Ackley's* functions. It can be observed from the figures that VMCSO preserves the fast-converging feature of the original CSO and accuracy and efficiency superiority. In fact, the VMCSO represents higher accelerations and higher volatility.

In Table IV, the numerical results for six test functions of the second bundle are shown. The best results among all the algorithms are shown in bold. For all the functions, the results show that VMCSO provides a better efficiency in lower swarm size while the other CSOs' performance depend on the swarm size significantly. The results are better with larger swarm size. It can be seen that VMCSO provides a better diversity with low swarm size, so it need less population size to get acceptable global best values and save the calculation time.

Fig. 3 shows the relationship between the global best values versus swarm sizes. Generally, VMCSO has the similar values for all the swarm sizes and provide a better efficiency in low swarm sizes.

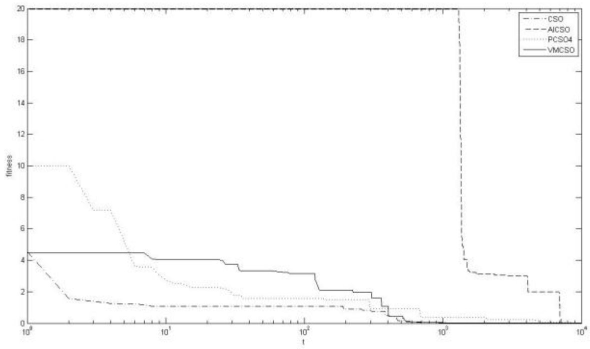

Fig. 2. Convergence graph on Ackley's function in 30 dimensions.

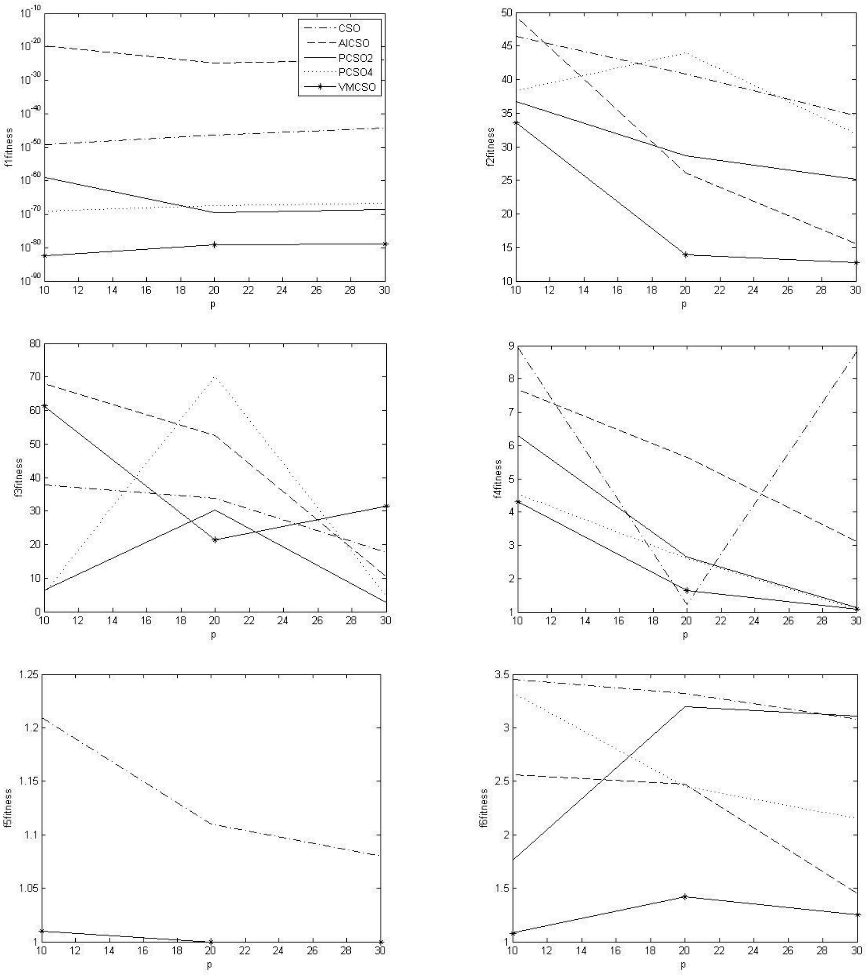

Fig. 3. Average best values versus swarm sizes for six functions in 30 dimensions.

IV. Conclusions

In this paper, we present a new VMCSO algorithm by adding a vibrational mutation operator in tracing mode to improve the performance of CSO in the tracing mode. VMCSO provide the cat swarm diversity during the optimization process.

It's easy to understand, implement and convenient to use. Experimental results show that VMCSO achieves better performance and more efficient and faster than other CSO variants methods investigated in this paper.

References

[1] S. Chu, P. W. Tsai, and J. S. Pan, *PRICAI 2006: Trends in Artificial Intelligence Lecture Notes in Computer Science*, Berlin, Germany: Springer Verlag, 2006, vol. 4099, pp. 854-858.

[2] S. Hackwood and G. Beni, "Self-organization of Sensors for Swarm Intelligence," in *Proc. the 1992 IEEE International Conference on Robotics and automatics*, 1992.

[3] B. Santosa and M. K. Ningrum, "Cat swarm optimization for clustering," in *Proc. the International Conference on Soft Computing and Pattern Recognition*, 2009, pp. 54-59.

[4] J. Paul and T. Yusiong, "Optimizing Artificial Neural Networks using Cat Swarm Optimization Algorithm," *I. J. Intelligent Systems and Applications*, vol. 1, pp. 69-80, 2013.

[5] S. Chittineni, K. Abhilash, V. Mounica, N. Sharada, and S. Satapathy, "Cat swarm optimization based neural network and particle swarm optimization based neural network in stock rates prediction," in *Proc. the 3rd International Conferences on Machine Learning and Computing*, 2011, pp. 292-296.

[6] P. P. Mohan and P. Ganapati, "Solving multiobjective problems using cat swarm optimization," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2956-2964, 2012.

[7] T. R. Benala and S. C. Satapathy, "Cat swarm optimisation for optimizing hybridized smoothening filter in image edge enhancement,"

in *Proc. the International Conferences on Syatemics, Cybernetics and Informatics*, 2010, pp. 247-252.

[8] J. Kennedy and R. Eberhart, "Partical swarm optimization," in *Proc. IEEE International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948.

[9] P. W. Tsai, J. S. Pan, S. M. Chen, B. Y. Liao, and S. P. Hao "Parallel Cat Swarm Optimization," in *Proc. 7th International Conference on Machine Learning and Cybernetics*, 2008, pp. 3328-3333.

[10] M. Orouskhani, M. Mansouri, and M. Teshnehlab, "Average-inertia weighted cat swarm optimization," in *Proc. 2nd International Conference on Swarm Intelligence*, 2011, pp. 321-328.

[11] S. Chu, "Computational intelligence based on the behavior of cats," *International Journal of Innovative Computing, Information and Control*, vol. 3, pp. 163-173, Feb., 2007.

[12] Y. V. Pehlivanoglu, "A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3. pp. 436-452, June, 2013.

[13] H. Pohlheim. Examples of Objective Functions. [Online]. Available: http://www.geatbx.com/download/GEATbx_ObjFunExpl_v37.pdf

[14] M. Molga and C. Smutnicki. (2005). Test functions for optimization needs. [Online]. Available: http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf

Yan Zhang received the B.S. degree in electronic information science and technology from Lanzhou University, Lanzhou Lanzhou, China, in 2011. She is currently persuing the M.S. degree in circuit and system at Lanzhou University. Her current research interests include biomedical engineering and evolutionary computation.

Yide Ma received the B.S. and M.S. degrees in radio technology from Chengdu University of Engineering Science and Technology, Chengdu, China, in 1984 and 1988, respectively. He received the Ph.D. degree from Lanzhou University, Lanzhou, China, in 2001.

He is currently a professor in the School of Information Science and Engineering, Lanzhou University. He has published more than 50 papers in major journals and international conferences and several textbooks, including *Principle and Application of Pulse Coupled Neural Network* (Beijing: Science Press, 2006), and *Principle And Application of Microcomputer* (Beijing: Science Press, 2006). His current research interests include artificial neural networks, digital image processing, pattern recognition, digital signal processing, and computer vision.