2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Maintaining Population Diversity in Brain Storm Optimization Algorithm

Shi Cheng*†, Yuhui Shi‡, Quande Qin§, T. O. Ting‡, and Ruibin Bai*
*Division of Computer Science, University of Nottingham Ningbo, China
†International Doctoral Innovation Centre, University of Nottingham Ningbo, China
‡Department of Electrical & Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China
§College of Management, Shenzhen University, Shenzhen, China
{shi.cheng, ruibin.bai}@nottingham.edu.cn, yuhui.shi@xjtlu.edu.cn

***Abstract*—Swarm intelligence suffers the premature convergence, which happens partially due to the solutions getting clustered together, and not diverging again. The brain storm optimization (BSO), which is a young and promising algorithm in swarm intelligence, is based on the collective behavior of human being, that is, the brainstorming process. Premature convergence also happens in the BSO algorithm. The solutions get clustered after a few iterations, which indicate that the population diversity decreases quickly during the search. A definition of population diversity in BSO algorithm to measure the change of solutions' distribution is proposed in this paper. The algorithm's exploration and exploitation ability can be measured based on the change of population diversity. Two kinds of partial re-initialization strategies are utilized to improve the population diversity in BSO algorithm. The experimental results show that the performance of the BSO is improved by these two strategies.**

***Index Terms*—Brain storm optimization, convergence, exploration/exploitation, population diversity, swarm intelligence.**

I. Introduction

Evolutionary computation algorithm is inspired from the natural selection process of the physical world, and the swarm intelligence mimics the behaviors of a population of animals/humans in the real world. Both evolutionary computation algorithms and swarm intelligence algorithms can be seen as decentralized systems, and a population of interacting individuals searches in the solution space to optimize a function or goal based on collective adaptation [1].

Brain storm optimization (BSO) algorithm is a new swarm intelligence algorithm, which mimics the brainstorming process in which a group of people solves a problem together [2], [3]. In a brain storm optimization algorithm, the solutions are divided into several clusters. The solutions being divided into several clusters can be seen as the population diverging into separate species, which are similar to the speciation in the natural selection. The new solutions are generated based on individual(s) in one or two clusters.

Optimization concerns with finding the "best available" solution(s) for a given problem. Optimization problems can be simply divided into unimodal problems and multimodal problems. As indicated by the name, a unimodal problem has only one optimum solution; on the contrary, a multimodal problem has several or numerous optimum solutions, of which many are local optimal solutions. Swarm intelligence (SI), and evolutionary algorithms (EA), are generally difficult to find global optimum solutions for multimodal problems due to the possible occurrence of premature convergence [4]–[6].

Swarm intelligence is based on a population of individuals [7]. In swarm intelligence, an algorithm maintains and successively improves a collection of potential solutions until some stopping condition is met. The solutions are initialized randomly in the search space, and are guided toward the better and better areas through the interaction among solutions.

In swarm intelligence algorithms, there are several solutions which exist at the same time. The premature convergence may happen due to the solution getting clustered together too fast. The population diversity is a measure of exploration and exploitation. Based on the population diversity changing measurement, the state of exploration and exploitation can be obtained. The population diversity definition is the first step to give an accurate observation of the search state. Many studies of population diversity in evolutionary computation algorithms and swarm intelligence have been proposed in [5], [8]–[14]

In this paper, we give a population diversity definition of the brain storm optimization algorithm, and propose two partial re-initializing solutions strategies to enhance the population diversity and to help solutions jump out of local optima. The idea behind the re-initialization is to increase the possibility for solutions "jumping out" of local optima, and to keep the ability for the algorithm to find "good enough" solution.

This paper is organized as follows. Section II reviews the basic brain storm optimization algorithm. Section III gives the definition of population diversity and the diversity maintaining strategies of BSO algorithm. Experiments on unimodal and multimodal benchmark functions are conducted in Section IV. The analysis and discussion of the performance of the BSO algorithm and the population diversity maintaining are given in Section V. Finally, Section VI concludes with some remarks and future research directions.

II. Brain Storm Optimization

The brain storm optimization, which is a young and promising algorithm in swarm intelligence, is based on the collective behavior of human being, that is, the brainstorming process [2], [3]. The speciation is a process of natural selection, which means that the population diverging into separate species

[15], [16]. The solutions in BSO are also diverging into several clusters. The new solutions are generated based on the mutation of one individual or interactive of two individuals.

The original BSO algorithm is simple in concept and easy in implementation. The main procedure is given in Algorithm 1. There are three strategies in this algorithm: the solution clustering, new individual generation, and selection [17].

The brain storm optimization algorithm is a kind of search space reduction algorithm [18]; all solutions will get into several clusters eventually. These clusters indicate a problem's local optima. The information of an area contains solutions with good fitness values are propagated from one cluster to another [19]. This algorithm will explore in decision space at first, and the exploration and exploitation will get into a state of equilibrium after iterations.

Algorithm 1: The procedure of the brain storm optimization algorithm

Initialization: Randomly generate n potential solutions (individuals), and evaluate the n individuals;
while *have not found "good enough" solution or not reached the pre-determined maximum number of iterations* **do**
  Clustering: Cluster n individuals into m clusters by a clustering algorithm;
  New individuals' generation: randomly select one or two cluster(s) to generate new individual;
  Selection: The newly generated individual is compared with the existing individual with the same individual index, the better one is kept and recorded as the new individual;
  Evaluate the n individuals;

The brain storm optimization algorithm also can be extended to solve multiobjective optimization problems [20]. Unlike the traditional multiobjective optimization methods, the brain storm optimization algorithm utilized the objective space information directly. Clusters are generated in the objective space; and for each objective, individuals are clustered in each iteration. The individual, which perform better in most of objectives are kept to the next iteration, and other individuals are randomly selected to keep the diversity of solutions.

A. Cluster Analysis

Cluster analysis is a kind of techniques that divides data into several groups (clusters). The goal of clustering is that objects being similar (or related) to one another are in the same cluster, and being different from (or related to) each other in different clusters.

Clustering is the process of grouping similar objects together. From the perspective of machine learning, the clustering analysis is sometimes termed as unsupervised learning. There are N points in the given input, $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$, the "interesting and/or useful pattern" can be obtained through the similarity calculation among points [21]. Every solution in the brain storm optimization algorithm is spread in the search space. The distribution of solutions can be utilized to reveal the landscapes of a problem.

Different clustering algorithms can be utilized in the brain storm optimization algorithm. In this paper, the basic k-means clustering algorithm is utilized.

B. Solution Clustering

The procedure of solution clustering is given in Algorithm 2. The clustering strategy divides individuals into several clusters. This strategy could refine a search area.

After many iterations, all solutions may be clustered into a small region. A probability value $p_{\text{clustering}}$ is utilized to control the probability of replacing a cluster center by a randomly generated solution. This could avoid the premature convergence, and help individuals "jump out" of the local optima.

Algorithm 2: The solution clustering strategy

Clustering: Cluster n individuals into m clusters by k-means clustering algorithm;
Rank individuals in each cluster and record the best individual as cluster center in each cluster;
Randomly generate a value $r_{\text{clustering}}$ in the range $[0, 1)$;
if *the value $r_{clustering}$ is smaller than a pre-determined probability $p_{clustering}$* **then**
  Randomly select a cluster center;
  Randomly generate an individual to replace the selected cluster center;

C. New Individual Generation

The procedure of new individual generation is given in Algorithm 3. A new individual can be generated based on one or several individuals or clusters. In the original brain storm optimization algorithm, a probability value $p_{\text{generation}}$ is utilized to determine a new individual being generated by one or two "old" individuals. Generating an individual from one cluster could refine a search region, and it enhances the exploitation ability. On the contrast, an individual, which is generated from two or more clusters, may be far from these clusters. The exploration ability is enhanced in this scenario.

The probability $p_{\text{oneCluster}}$ and probability $p_{\text{twoCluster}}$ are utilized to determine the cluster center or random individual will be chosen in one cluster or two clusters generation case, respectively. In one cluster generation case, the new individual from center or random individual can control the exploitation region. While in several clusters generation case, the random individuals could increase the population diversity of swarm.

The new individuals are generated according to the functions (1) and (2).

$$x_{\text{new}}^{i} = x_{\text{old}}^{i} + \xi(t) \times \text{rand}() \tag{1}$$

$$\xi(t) = \text{logsig}\left(\frac{0.5 \times T - t}{c}\right) \times \text{rand}() \tag{2}$$

Algorithm 3: The new individual generation strategy

New individual generation: randomly select one or two cluster(s) to generate new individual;
Randomly generate a value $r_{\text{generation}}$ in the range $[0, 1)$;
if *the value $r_{generation}$ is less than a probability $p_{generation}$* **then**
    Randomly select a cluster, and generate a random value $r_{\text{oneCluster}}$ in the range $[0, 1)$;
    if *the value $r_{oneCluster}$ is smaller than a pre-determined probability $p_{oneCluster}$;*
    then
        Select the cluster center and add random values to it to generate new individual;
    else
        Randomly select an individual from this cluster and add random value to the individual to generate new individual;
else
    randomly select two clusters to generate new individual;
    Generate a random value $r_{\text{towCluster}}$ in the range $[0, 1)$;
    if *the value $r_{towCluster}$ is less than a pre-determined probability $p_{twoCluster}$* **then**
        the two cluster centers are combined and then added with random values to generate new individual;
    else
        two individuals from each selected cluster are randomly selected to be combined and added with random values to generate new individual;
The newly generated individual is compared with the existing individual with the same individual index, the better one is kept and recorded as the new individual;

where x^i_{new} and x^i_{old} are the ith dimension of x_{new} and x_{old}; and the value x_{old} is a copy of one individual or the combination of two individuals. The parameter T is the maximum number of iterations, t is the current iteration number, c is a coefficient to change logsig() function's slope.

D. Selection

The selection strategy is utilized to keep good solutions in all individuals. A modified step size and individual generation was proposed in [22]. The step size can be utilized to balance the convergence speed of the algorithm. The better solutions are kept by the selection strategy, while clustering strategy and generation strategy add new solutions into the swarm to keep the diversity for the whole population.

III. Population Diversity

The most important factor affecting an optimization algorithm's performance is its ability of "exploration" and "exploitation." Exploration means the ability of a search algorithm to explore different areas of the search space in order to have high probability to find good promising solutions. Exploitation, on the other hand, means the ability to concentrate the search around a promising region in order to refine a candidate solution. A good optimization algorithm should optimally balance the two conflicted objectives [19], [23].

In a brain storm optimization algorithm, the solutions are grouped into several clusters. The best solutions of each cluster are kept to the next iteration due to the selection operation. New individual can be generated based on one or two individuals in clusters. The exploitation ability is enhanced when the new individual is close to the best solution so far. While the exploration ability is enhanced when the new individual is randomly generated, or generated by individuals in two clusters.

Population diversity is useful for measuring and dynamically adjusting an algorithm's ability of exploration or exploitation accordingly. In the brain storm optimization algorithm, many solutions are existed at the same time, and these solutions are gathered into several clusters. The solutions may get together into a small region after iterations. The clustering algorithm is difficult to cluster solutions into different group when every solution is within a small region. The algorithm's exploration ability is decreased at this time.

It is important to find a metric to measure the population diversity of solutions in the brain storm optimization algorithm. From the measurement, we can monitor the search of solutions.

A. Population Diversity Definition

Population diversity is a measurement of solutions' distribution. In [3], proposed D_c, D_v, and D_e to measure normalized distance for a cluster, inter-cluster diversity, and information entropy for the population, respectively. Here, in this paper, we define the population diversity given below, which is dimensional-wise and based on the L_1 norm.

$$\bar{x}_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij} \qquad D_j = \frac{1}{m}\sum_{i=1}^{m} |x_{ij} - \bar{x}_j|$$

$$D = \sum_{j=1}^{n} w_j D_j$$

where $\bar{x}_j$ represents the pivot of solutions in dimension j, and D_j measures solution diversity based on L_1 norm for dimension j. Then we define $\bar{\mathbf{x}} = [\bar{x}_1, \cdots, \bar{x}_j, \cdots, \bar{x}_n]$, $\bar{\mathbf{x}}$

represents the mean of current solutions on each dimension, and $\mathbf{D} = [D_1, \cdots, D_j, \cdots, D_n]$, which measures solution diversity based on L_1 norm for each dimension. D measures the whole group's population diversity.

Without loss of generality, every dimension is considered equally. Setting all weights $w_j = \frac{1}{n}$, then the dimension-wise position diversity can be rewritten as:

$$D^p = \sum_{j=1}^{n} \frac{1}{n} D_j^p = \frac{1}{n} \sum_{j=1}^{n} D_j^p$$

B. Maintaining Population Diversity

Population diversity is a measurement of population state of exploration or exploitation. It illustrates the distribution of solutions. The solutions diverging means that the search is in an exploration state, on the contrary, solutions clustering tightly means that the search is in an exploitation state [24].

The solutions get clustered in search space, and it may not be easy to diverge. The population diversity is decreased when all solutions are clustered into one small region. Many strategies are proposed to enhance the population diversity in evolutionary computation algorithms and swarm intelligence. These strategies include inserting randomly generated individuals, niching [25], [26], solutions re-initialization [18], [24], or reconstructing the fitness function with the consideration of the age of individuals [27] or the entropy of the population [28].

In this paper, the solutions partial re-initialization is utilized to promote diversity of BSO algorithm. In the brain storm optimization algorithm, the new individual is generated by adding one or two individual(s) with the noise based on equation (1). However, every solution will be very similar in each dimension when the solutions get clustered into a small region. The original BSO algorithm may not be easy to escape from local optima. The partial re-initialization in the whole search space could make many solutions diverge into large search areas. The idea behind the re-initialization is to increase possibility for solutions "jumping out" of local optima, and to keep the ability for algorithm to find "good enough" solutions.

Algorithm 4 gives the procedure of the BSO algorithm with re-initialization strategy. After several iterations, part of solutions re-initializes its position and velocity in whole search space, which increases the possibility of solutions "jumping out" of local optima. According to the number of re-initialized solutions, this strategy can be divided into two kinds.

- Half solutions re-initialized after certain iterations. This approach can obtain a great ability of exploration due to the possibility that half of solutions will have the chance to escape from local optima.
- The number of re-initialized solutions is decreasing during the search process. More than half solutions are re-initialized at the beginning of search, and the number of re-initialized solutions is linearly decreased at each re-initialization. This strategy is to focus on the exploration at first, and the exploitation at the end of the search.

Algorithm 4: The procedure of the population diversity promoted BSO algorithm

```
Initialization: Randomly generate n potential solutions
(individuals), and evaluate the n individuals;
while have not found "good enough" solution or not
reached the pre-determined maximum number of
iterations do
    Clustering: Cluster n individuals into m clusters by a
    clustering algorithm;
    New individual generation: randomly select one or
    two cluster(s) to generate new individual;
    Selection: The newly generated individual is
    compared with the existing individual with the same
    individual index, the better one is kept and recorded
    as the new individual;
    Re-initialization: partially re-initialize some
    solutions after certain iterations;
    Evaluate the n individuals;
```

IV. Experimental Study

Wolpert and Macerady have proved that under certain assumptions no algorithm is better than other one on average for all problems [29]. The aim of the experiment is not to compare the ability or the efficacy of the brain storm optimization algorithm with other swarm intelligence algorithms, but the population diversity property of the brain storm optimization algorithm.

A. Benchmark Test Functions and Parameter Setting

The experiments have been conducted to test the proposed BSO algorithm on the benchmark functions listed in Table I. Considering the generality, eleven standard benchmark functions were selected, which include five unimodal functions and seven multimodal functions [30], [31]. All functions are run 50 times to ensure a reasonable statistical result. There are 1500 iterations for 50 dimensional problems in every run. Randomly shifting of the location of optimum is utilized in each dimension for each run.

In all experiments, the brain storm optimization has 200 individuals, and parameters are set as the following, let $p_{\text{clustering}} = 0.2$, $p_{\text{generation}} = 0.6$, $p_{\text{oneCluster}} = 0.4$ and $p_{\text{twoCluster}} = 0.5$. The parameter k in k-means algorithm is 20. The coefficient c is set as 20.0. In the BSO with solution re-initialization, the solutions will be partially re-initialized after each 200 iterations. In the decreasing number of solution re-initialization case, there are 20 solutions are kept at the first time, the number of kept solutions increase 20 at each re-initialization, and 140 solutions are kept at the last time.

B. Experimental Results

Several measures of performance are utilized in this paper. The first is the best fitness value attained after a fixed number of iterations. In our case, we report the best result found after 1500 for 50 dimensional problems. The following measures

TABLE I
THE BENCHMARK FUNCTIONS USED IN EXPERIMENTAL STUDY, WHERE n IS THE DIMENSION OF EACH PROBLEM, $= (\mathbf{x} - \mathbf{o})$, $\mathbf{x} = [x_1, x_2, \cdots, x_n]$, o_i IS AN RANDOMLY GENERATED NUMBER IN PROBLEM'S SEARCH SPACE S AND IT IS DIFFERENT IN EACH DIMENSION, GLOBAL OPTIMUM $\mathbf{x}^* = \mathbf{o}$, $f_{\min}$ IS THE MINIMUM VALUE OF THE FUNCTION, AND $S \subseteq \mathcal{R}^n$.

Function	Test Function	S	$f_{\min}$
Parabolic	$f_0(\mathbf{x}) = \sum_{i=1}^{n} z_i^2 + \text{bias}_0$	$[-100, 100]^n$	-450.0
Schwefel's P2.22	$f_1(\mathbf{x}) = \sum_{i=1}^{n} \lvert z_i\rvert + \prod_{i=1}^{n} \lvert z_i\rvert + \text{bias}_1$	$[-10, 10]^n$	-330.0
Schwefel's P1.2	$f_2(\mathbf{x}) = \sum_{i=1}^{n} (\sum_{k=1}^{i} z_k)^2 + \text{bias}_2$	$[-100, 100]^n$	450.0
Step	$f_3(\mathbf{x}) = \sum_{i=1}^{n} (\lfloor z_i + 0.5 \rfloor)^2 + \text{bias}_3$	$[-100, 100]^n$	330.0
Quartic Noise	$f_4(\mathbf{x}) = \sum_{i=1}^{n} i z_i^4 + \text{random}[0, 1) + \text{bias}_4$	$[-1.28, 1.28]^n$	-450.0
Rosenbrock	$f_5(\mathbf{x}) = \sum_{i=1}^{n-1} [100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2] + \text{bias}_5$	$[-10, 10]^n$	180.0
Rastrigin	$f_6(\mathbf{x}) = \sum_{i=1}^{n} [z_i^2 - 10\cos(2\pi z_i) + 10] + \text{bias}_6$	$[-5.12, 5.12]^n$	-330.0
Noncontinuous Rastrigin	$f_7(\mathbf{x}) = \sum_{i=1}^{n} [y_i^2 - 10\cos(2\pi y_i) + 10] + \text{bias}_7$ $y_i = \begin{cases} z_i & \lvert z_i\rvert < \frac{1}{2} \\ \frac{\text{round}(2z_i)}{2} & \lvert z_i\rvert \geq \frac{1}{2} \end{cases}$	$[-5.12, 5.12]^n$	450.0
Ackley	$f_8(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)\right) + 20 + e + \text{bias}_8$	$[-32, 32]^n$	180.0
Griewank	$f_9(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n} z_i^2 - \prod_{i=1}^{n}\cos(\frac{z_i}{\sqrt{i}}) + 1 + \text{bias}_9$	$[-600, 600]^n$	120.0
Generalized Penalized	$f_{10}(\mathbf{x}) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2$ $\times[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+\sum_{i=1}^{n} u(z_i, 10, 100, 4) + \text{bias}_{10}$ $y_i = 1 + \frac{1}{4}(z_i + 1)$ $u(z_i, a, k, m) = \begin{cases} k(z_i - a)^m & z_i > a, \\ 0 & -a < z_i < a \\ k(-z_i - a)^m & z_i < -a \end{cases}$	$[-50, 50]^n$	330.0

are the median, the worst and mean value of best fitness values in each run. It is possible that an algorithm will rapidly reach a relatively good result while becoming trapped into a local optimum. These three values give a measure of algorithms' reliability and robustness.

Table II gives results of the brain storm optimization algorithm solving unimodal and multimodal problems. The population diversity enhanced BSO performs better than the original BSO for most problems, especially for the unimodal problems.

For traditional algorithms, the multimodal problems are difficult to solve than unimodal problems due to that the multimodal problems have many local optima. However, the brain storm optimization algorithm may be more suitable for multimodal problems. The concept of brain storm optimization algorithm is not to cluster all solutions into one small region, but many regions. From the results, we can find that the original BSO algorithm performs well on the multimodal functions, and the population diversity enhanced BSO algorithm have more improvement in solving unimodal functions than multimodal functions.

V. Analysis and Discussion

A. Population Diversity Monitor

Due to the limit of space, the simulation results of six representative benchmark functions are reported here. The functions include three unimodal and three multimodal problems. The unimodal functions include Parabolic f_0, Schwefel's P1.2 f_2, Step f_3, and the multimodal functions include Generalized Rosenbrock f_5, Noncontinuous Rastrigin f_7, and Generalized Penalized f_{10}.

Figure 1 displays the average performance of brain storm optimization algorithms solving three unimodal and three multimodal functions. The brain storm optimization algorithm has a fast convergence at the beginning of search, which indicates that the good search regions can be located after several solution clustering strategies. However, the ability of preventing premature convergence, and "jumping out" of local

TABLE II

RESULT OF BRAIN STORM OPTIMIZATION SOLVING UNIMODAL AND MULTIMODAL BENCHMARK FUNCTIONS. ALL ALGORITHMS ARE RUN FOR 50 TIMES, WHERE "BEST", "MEDIAN", "WORST", AND "MEAN" INDICATE THE BEST, MEDIAN, WORST, AND MEAN OF THE BEST FITNESS VALUES FOR ALL RUNS, RESPECTIVELY.

Func.	$f_{\min}$	Dim	Best	Median	Worst	Mean	Std. Dev.
		original	-283.7674	69.3182	1011.455	128.2867	268.266
f_0	-450.0	half	**-413.8930**	-292.6541	-73.4257	**-282.6678**	78.2084
		decrease	-389.3168	-296.8813	89.42494	-279.8117	93.4272
		original	-329.9999	-329.9987	-329.0885	-329.9615	0.16351
f_1	-330.0	half	**-329.9999**	-329.9975	-329.9844	-329.9968	0.00311
		decrease	-329.9999	-329.9978	-329.9872	**-329.9969**	0.00297
		original	1674.185	3469.0662	6521.9105	3715.998	1155.126
f_2	450.0	half	1236.369	1715.393	2518.608	1770.651	365.281
		decrease	**1013.162**	1734.731	2432.571	**1682.709**	311.921
		original	1461	1989	4036	2121.96	450.6883
f_3	330.0	half	**765**	1122	1548	1110.22	163.962
		decrease	785	1095	1536	**1086.92**	173.0487
		original	**-449.9989**	-449.9966	-449.9933	**-449.9963**	0.00116
f_4	-450.0	half	-449.9983	-449.9960	-449.9934	-449.9960	0.00114
		decrease	-449.9978	-449.9955	-449.9922	-449.9955	0.00130
		original	221.5290	227.5745	288.2411	**232.1447**	15.4617
f_5	180.0	half	**219.0803**	227.5494	389.8904	239.6247	33.4026
		decrease	221.2358	227.3915	360.8707	237.8019	26.1117
		original	-300.1512	-265.3277	-224.5344	-264.9098	17.7013
f_6	-330.0	half	**-301.1461**	-271.2974	-198.6657	**-267.5166**	20.5183
		decrease	-295.1764	-263.3378	-190.7060	-260.3728	22.7473
		original	**482**	526	619	528.68	27.3447
f_7	450.0	half	487	532	592	**528.04**	20.8891
		decrease	487	528	606	530.58	25.4684
		original	188.2361	190.7374	192.2957	190.6498	0.84468
f_8	180.0	half	**186.9072**	190.3279	192.1137	**190.1500**	1.13627
		decrease	187.4559	190.5704	191.8832	190.2716	1.07905
		original	129.8876	134.4022	142.2110	134.6174	2.88916
f_9	120.0	half	**124.2548**	126.2922	130.5381	126.3547	1.28784
		decrease	124.3876	126.1242	129.6540	**126.1661**	1.14855
		original	**332.1512**	336.5663	344.5289	337.1611	2.89567
f_{10}	330.0	half	332.5938	336.9778	344.0822	337.3973	2.97637
		decrease	332.1948	336.2336	345.7965	**337.0947**	2.84417

optima should be improved. Keeping the global search ability, and improving the local search ability should be investigated in the brain storm optimization algorithm.

B. Population Diversity Analysis

Figure 2 displays the population diversity changes during the search process. There are many vibrations of population diversity change in the original BSO solving unimodal functions. The population diversity changes smoothly in the original BSO solving multimodal functions. This may be caused by the different properties of BSO solving unimodal and multimodal functions, and more investigation should be taken on the mechanism of BSO solving different types of problems.

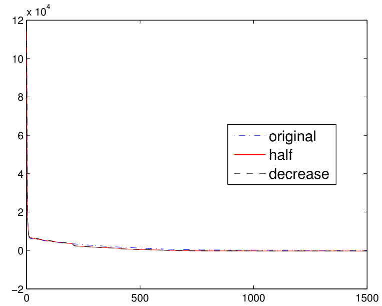
The population diversity maintained BSO has promoted the population diversity after certain iterations. The value of population diversity is kept at a large number during the search, this could help the solutions "jump out" a local optima.

VI. CONCLUSION

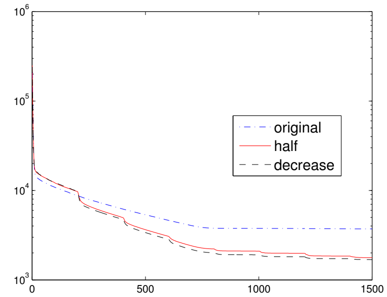
In swarm intelligence algorithms, premature convergence happens partially due to the solutions getting clustered together, and not diverging again. The premature convergence also happens in the brain storm optimization algorithm. To prevent the premature convergence, algorithm's exploration ability and exploitation ability should be balanced during the search.

The population diversity is a measure of exploration and exploitation. Based on the population diversity changing measurement, the state of exploration and exploitation can be obtained. The population diversity definition is the first step to give an accurate observation of the search state. Many approaches have been introduced based on the idea that prevents solutions from clustering too tightly in one region of the search space to achieve great possibility to "jump out" of local optima [32].

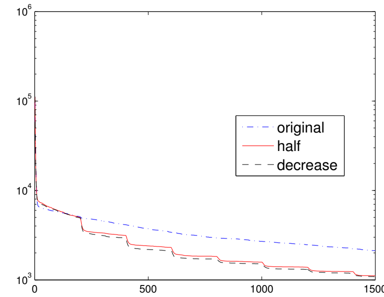
In this paper, we give a population diversity definition of the brain storm optimization algorithm, and propose two kinds of diversity enhanced strategies to help solutions jump out of local optima. The experimental study shows that the performance of optimization is improved by the population diversity enhancement. The population diversity also should be



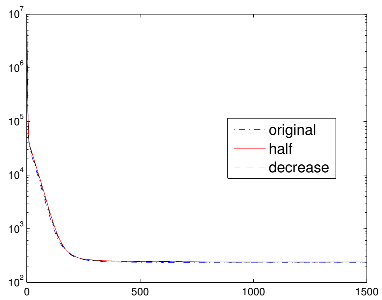
(a) Parabolic f_0



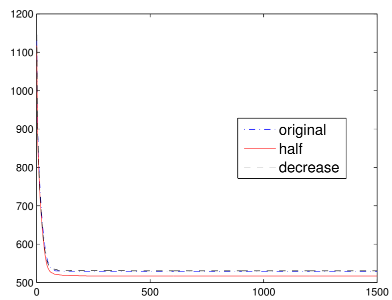
(b) Schwefel's P1.2 f_2



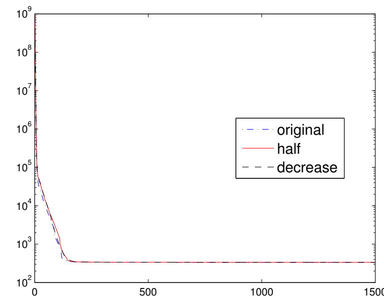
(c) Step f_3



(d) Rosenbrock f_5

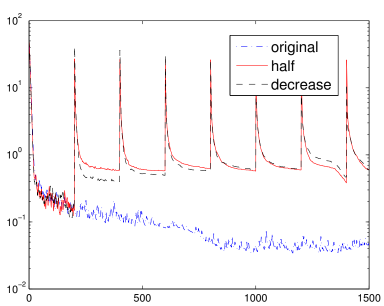


(e) Noncontinuous Rastrigin f_7

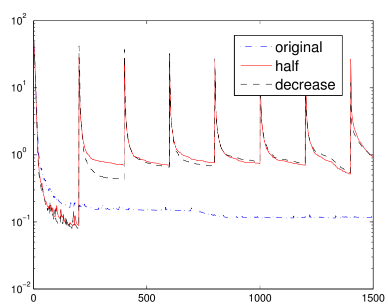


(f) Generalized Penalized f_{10}

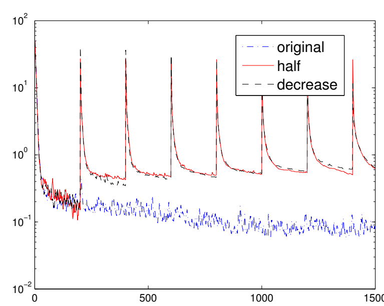
Fig. 1. The average performance of the brain storm optimization algorithm solving unimodal and multimodal functions. The unimodal functions are (a) Parabolic f_0, (b) Schwefel's P1.2 f_2, and (c) Step f_3; the multimodal functions are (d) Rosenbrock f_5, (e) Noncontinuous Rastrigin f_7, and (f) Generalized Penalized f_{10}.



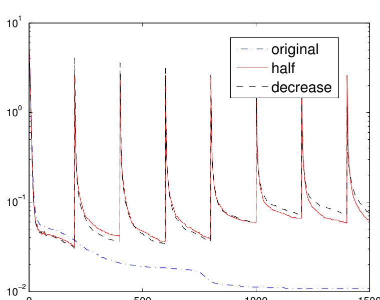
(a) Parabolic f_0



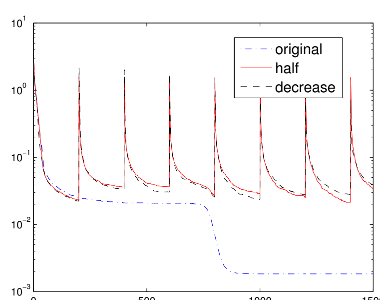
(b) Schwefel's P1.2 f_2



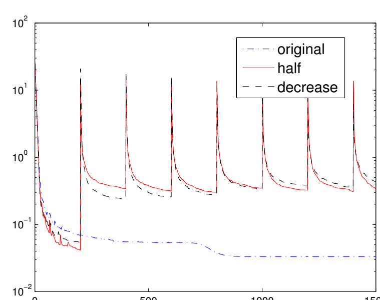
(c) Step f_3



(d) Rosenbrock f_5



(e) Noncontinuous Rastrigin f_7



(f) Generalized Penalized f_{10}

Fig. 2. The population diversity monitor of the brain storm optimization algorithm solving unimodal and multimodal functions. The unimodal functions are (a) Parabolic f_0, (b) Schwefel's P1.2 f_2, and (c) Step f_3; the multimodal functions are (d) Rosenbrock f_5, (e) Noncontinuous Rastrigin f_7, and (f) Generalized Penalized f_{10}.

monitored in the brain storm optimization algorithm solving multiobjective problems. The relationship between the population diversity changes and the performance of BSO algorithm, and the properties of population diversity changes with different problems also needs more analysis. In general, the brain storm optimization algorithm is a young and promising algorithm; there are many fields which are under investigation.

ACKNOWLEDGMENT

This work was carried out at the International Doctoral Innovation Centre (IDIC). The authors acknowledge the financial support from Ningbo Education Bureau, Ningbo Science and Technology Bureau, China's MOST and The University of Nottingham. This work is also partially supported by National Natural Science Foundation of China under grant No.60975080, 61273367; and Ningbo Science & Technology Bureau (Science and Technology Project No.2012B10055).

REFERENCES

[1] A. Engelbrecht, X. Li, M. Middendorf, and L. M. Gambardella, "Editorial special issue: Swarm intelligence," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 677–680, August 2009.

[2] Y. Shi, "Brain storm optimization algorithm," in *Advances in Swarm Intelligence*, ser. Lecture Notes in Computer Science, Y. Tan, Y. Shi, Y. Chai, and G. Wang, Eds. Springer Berlin/Heidelberg, 2011, vol. 6728, pp. 303–309.

[3] ——, "An optimization algorithm based on brainstorming process," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 2, no. 4, pp. 35–62, October-December 2011.

[4] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, Department of Computer and Communication Sciences, University of Michigan, August 1975.

[5] M. L. Mauldin, "Maintaining diversity in genetic search," in *Proceedings of the National Conference on Artificial Intelligence (AAAI 1984)*, August 1984, pp. 247–250.

[6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[7] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*. Morgan Kaufmann Publisher, 2001.

[8] E. K. Burke, S. Gustafson, and G. Kendall, "A survey and analysis of diversity measures in genetic programming," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 716–723.

[9] Y. Shi and R. Eberhart, "Population diversity of particle swarms," in *Proceedings of the 2008 Congress on Evolutionary Computation (CEC2008)*, 2008, pp. 1063–1067.

[10] ——, "Monitoring of particle swarm optimization," *Frontiers of Computer Science*, vol. 3, no. 1, pp. 31–37, March 2009.

[11] S. Cheng and Y. Shi, "Diversity control in particle swarm optimization," in *Proceedings of 2011 IEEE Symposium on Swarm Intelligence (SIS 2011)*, Paris, France, April 2011, pp. 110–118.

[12] S. Cheng, Y. Shi, and Q. Qin, "Experimental Study on Boundary Constraints Handling in Particle Swarm Optimization: From Population Diversity Perspective," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 2, no. 3, pp. 43–69, July-September 2011.

[13] S. Cheng, "Population diversity in particle swarm optimization: Definition, observation, control, and application," Ph.D. dissertation, Department of Electrical Engineering and Electronics, University of Liverpool, May 2013.

[14] S. Cheng, Y. Shi, and Q. Qin, "A study of normalized population diversity in particle swarm optimization," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 4, no. 1, pp. 1–34, January-March 2013.

[15] C. Darwin, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, 5th ed. London: John Murray, 1869.

[16] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham, *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*, ser. Numerical Insights, A. Sydow, Ed. Chapman & Hall/CRC Press, 2009, vol. 6.

[17] S. Cheng, Y. Shi, Q. Qin, and S. Gao, "Solution clustering analysis in brain storm optimization algorithm," in *Proceedings of The 2013 IEEE Symposium on Swarm Intelligence, (SIS 2013)*. Singapore: IEEE, 2013, pp. 111–118.

[18] S. Cheng, Y. Shi, and Q. Qin, "Dynamical exploitation space reduction in particle swarm optimization for solving large scale problems," in *Proceedings of 2012 IEEE Congress on Evolutionary Computation, (CEC 2012)*. Brisbane, Australia: IEEE, 2012, pp. 3030–3037.

[19] ——, "Population diversity based study on search information propagation in particle swarm optimization," in *Proceedings of 2012 IEEE Congress on Evolutionary Computation, (CEC 2012)*. Brisbane, Australia: IEEE, 2012, pp. 1272–1279.

[20] J. Xue, Y. Wu, Y. Shi, and S. Cheng, "Brain storm optimization algorithm for multi-objective optimization problems," in *Advances in Swarm Intelligence*, ser. Lecture Notes in Computer Science, Y. Tan, Y. Shi, and Z. Ji, Eds. Springer Berlin / Heidelberg, 2012, vol. 7331, pp. 513–519.

[21] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, ser. Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, 2012.

[22] D. Zhou, Y. Shi, and S. Cheng, "Brain storm optimization algorithm with modified step-size and individual generation," in *Advances in Swarm Intelligence*, ser. Lecture Notes in Computer Science, Y. Tan, Y. Shi, and Z. Ji, Eds. Springer Berlin / Heidelberg, 2012, vol. 7331, pp. 243–252.

[23] S. Cheng, Y. Shi, and Q. Qin, "Population diversity of particle swarm optimizer solving single and multi-objective problems," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 3, no. 4, pp. 23–60, 2012.

[24] ——, "Promoting diversity in particle swarm optimization to solve multimodal problems," in *Neural Information Processing*, ser. Lecture Notes in Computer Science, B.-L. Lu, L. Zhang, and J. Kwok, Eds. Springer Berlin / Heidelberg, 2011, vol. 7063, pp. 228–237.

[25] W. Cedeño and V. R. Vemuri, "On the use of niching for dynamic landscapes," in *Proceedings of 1997 IEEE Congress on Evolutionary Computation, (CEC 1997)*. IEEE, 1997, pp. 361–366.

[26] A. Della Cioppa, C. De Stefano, and A. Marcelli, "Where are the niches? dynamic fitness sharing," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 4, pp. 453–465, August 2007.

[27] A. Ghosh, S. Tsutsui, and H. Tanaka, "Function optimization in nonstationary environment using steady state genetic algorithms with aging of individuals," in *Proceedings of 1998 IEEE Congress on Evolutionary Computation, (CEC 1998)*. IEEE, 1998, pp. 666–671.

[28] Y. Jin and B. Sendhoff, "Constructing dynamic optimization test problems using the multi-objective optimization concept," in *Applications of Evolutionary Computing*, ser. Lecture Notes in Computer Science, G. R. Raidl, S. Cagnoni, J. Branke, D. W. Corne, R. Drechsler, Y. Jin, C. G. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G. D. Smith, and G. Squillero, Eds. Springer Berlin / Heidelberg, 2004, vol. 3005, pp. 525–536.

[29] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997.

[30] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, July 1999.

[31] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, June 2006.

[32] T. Blackwell and P. Bentley, "Don't push me! collision-avoiding swarms," in *Proceedings of The Fourth Congress on Evolutionary Computation (CEC 2002)*, May 2002, pp. 1691–1696.