Benchmark Learning Algorithm

Anshi Xie

School of Public Policy and Management, Tsinghua University, Beijing 100084, China

shermanxas@163.com

***Abstract*-In this paper, the benchmarking learning algorithm (BLA), was proposed according to the benchmark learning theory in the business management. In BLA, a competitive learning mechanism based on dynamic niche was set up. First, by right of imitation and learning, all the individuals within population were able to approach to the high yielding regions in the solution space, and seek out the optimal solutions quickly. What is more, the premature convergence problem was solved through new optimal solution policy. Last but not least, BLA is able to accurately detect the slight changes of the environments and track the trajectory of the extreme points in the search space. And thus, it is naturally adaptable for the dynamic optimization problems. In this paper, the main differences between BLA and the existing intelligent optimization methods, such as genetic algorithm (GA), particle swarm optimization (PSO) et al were analyzed and revealed. The comparative experiments for both the static optimization problem and the dynamic optimization problem showed that BLA is robust and able to perform friendly interactive learning with the environments, whose search speed, optimization ability and dynamic tracking ability were far superior to other similar methods.**

Keywords- Benchmark Learning; Search Pattern; Evolutionary Algorithm; Swarm Intelligence; Dynamic Environments

I. INTRODUCTION

Intelligent computation, also known as natural computation, is type of optimization model inspired by the principles of natural world, especially the biological world. Many optimization algorithms are included in of intelligent computation, which mainly consists of Evolutionary Algorithms (EAs) and Swarm Intelligences (SIs) et al. EAs include four main branches, such as Genetic Algorithm (GA), Evolutionary Programming (EP), Evolutionary Strategy (ES), Genetic Programming (GP). SIs include Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Artificial Fish Swarm Algorithm (AFSA), and Shuffled Frog Leaping Algorithm (SFLA). Besides EAs and SIs, Simulated Annealing (SA), Tabu Search (TS) and Predatory Search (PS) should be included within intelligent computation as well.

In 1975, professor Holland from the United States proposed the classic Genetic Algorithm [1] (GA) for the first time. Kirkpatrick introduced the ideas of simulated annealing to the optimization research field, then Simulated Annealing algorithm [2] (SA) is proposed in 1982. Micro-canonical Annealing [3] (MA) was raised by professor Creutz in 1983, which is similar to SA, but a deterministic way, instead of the Metropolis criterion, was involved in the process of state transition in annealing. Now the existing research about MA was mainly used in the field of image processing. Memetic Algorithm [4] (MA) was put forward for the first time by Pablo Moscato in 1989. In 1991, Alberto and Dorigo from Italy proposed the basic model of Ant Colony Optimization [5] (ACO). In 1995, Dr Kennedy and his partner, Eberhart, proposed the Particle Swarm Optimization algorithm [6] (PSO) according to the birds feeding behavior model. In 1997, Rainer Storn and Kenneth Price presented Differential Evolution algorithm [7] (DE) based on the ideal of GA. Castro from Brazil summarized the ideal of Artificial Immune System [8] (AIS) for the first time in 1999, and then, the famous Clonal Selection Algorithm [9] (CSA) was proposed based on the principle of clonal selection. Since entering the new century, all kinds of new algorithm have emerged. Three new algorithms appeared in 2000, like Eusuff and Lansey raised a new recta-heuristic Shuffled Frog leaping algorithm [10] (SFLA) and some improvements later; in the same year, Murase from Japan put forward a Photosynthetic Algorithm [11] (PA) based on the principle of plant photosynthesis, and it is useful for the N queens problem and finite element analysis; in the same year, Zelinka and Lampinen proposed the Self-Organizing Migrating Algorithm [12] (SOMA) based on the strategy of cooperation and competition among species. In 2001, Geem etc. raised a Harmony Search [13] (HS) in line with the harmony principle of music performance. In 2002, Passino put forward Bacteria Foraging Optimization Algorithm [14] (BFOA) according to the foraging behavior of escherichia coli. In the same year, Xie etc. proposed a Social Cognitive Optimization [15] (SCO) on the basis of social cognitive theory [16]. In 2003, Li XL presented Artificial Fish Swarm Algorithm [17] (AFSA) in the light of fish behavior patterns; in the same year, Zhou etc. put ward a kind of Population Migration Algorithm [18] (PMA) according to the principle of population migration. In 2004, Nakrani proposed the Bee Algorithm [19] (BA) based on the foraging behavior of bees for the first time, and then later, Karabog come up with the famous Artificial Bee Colony algorithm [20] (ABC) according to the social behavior model of the bee swarm. Three new algorithms emerged in 2005. Yang from the University of Cambridge proposed a Enzyme Algorithm [21] (EA) based on the principle of inhibition and catalysis of enzyme; Krishnanand and Ghose raised the Glowworm Swarm Optimization [22] (GSO) in accordance with the courtship behavior of the fire fly; LI etc. come up with Plant Growth Simulation Algorithm [23] (PGSA) on the basis of the plant phototropism mechanism. In 2007, Mucherino and Seref presented a Monkey Search [24] (MS) based on the social pattern of monkeys. Four new algorithms were proposed in 2008. Havens presented a kind of Roach Infestation Optimization [25] (RIO) on the basis of social behaviors of cockroaches; Yang presented a kind of Firefly Algorithm [26] (FA); Simon put forward a kind of Biogeography-based optimization [27] (BBO) according to the mathematical model of migratory species. In accordance with the principles of optics, especially the fermat principle, a Light Ray Optimization (LRO) was proposed by anonymous Chinese scholars for the first time in 2008, and then later, it was improved by some scholars and put to application [28]. MA etc. presented a kind of evolutionary algorithm [29] based on species migration optimization in 2009. In 2010, Tero etc. from Japan reported that a kind of slime mold formed the simulating real-world infrastructure networks with comparable efficiency, fault tolerance, and cost. And this study showed some characteristics of bionic calculation [30]. In 2011, four new algorithms were proposed by Chinese scholars, including the Physicomimetics Method [31] (PM) for Global Optimization, Explosion Search Algorithm [32] (ESA), Cell Membrane Optimization [33] (CMO) and Fruit Fly Optimization Algorithm [34, 35] (FOA).

These intelligent optimization algorithms mentioned above have special distinguishing features from each other, but they all have some common deficiencies. First of all, they all try to search the optimal solution by the individual's random drift in the solution space, yet the search direction and search purpose of the random drift are indeterminate and uncertain. Furthermore, they are all population convergence-oriented. That is to say, at the end of the search process, all the individuals are apt to converge to a certain point within the solution space. This point should have been the global optimal solution. However, all the individuals are likely to converge to a local optimal solution, because the population diversity cannot be maintained due to the convergence strategy. Finally, they are all designed and studied for static optimization problems and their search process are always passively adaptable, yet most of practical applications are dynamic and changeable. Since they are unable to maintain the population diversity, and unable to detect and make quick responses to the slight changes in the environments, they always loss the adaptabilities for the environments and cannot track the trajectories of the extreme value point in the solution space, and cannot adapt to solving dynamic optimization problems.

Benchmarking [36] originally means that a surveyor's mark on a permanent object of predetermined position and elevation used as a reference point. As a management idea and management method, benchmarking learning originated from enterprise management domain, and it means that some outstanding enterprise can be set as a standard, by which other companies can be measured or judged, and improved consequently.

Based on benchmarking concept, the benchmarking learning algorithm (BLA), as a new type of optimization model, was proposed in this paper, which is different from all the existing optimization methods, yet it is able to overcome their deficiencies. In BLA, according to the core values of benchmarking, namely comprehensive quality, process, standard and learning, a dynamic learning mechanism based on ecological niche technique was designed to conquer the disadvantages mentioned above.

II. THE BENCHMARKING LEARNING ALGORITHM (BLA)

Benchmarking learning, in short, is to find the best case and learn something from it via imitation, others will improve themselves and even beyond the opponent. So the main framework of BLA is that the entire ecological system, namely the whole solution space, consists of many niche populations, which are similar to the enterprise legal entities on the global market. And all the individuals within each niche population are similar to all the employees inside each enterprise. Both the best individual in each niche population, known as the local optimal individual, that is, the internal benchmarking, and the best individual in the whole ecological system, known as the global optimal individual, namely, the external benchmarking, were established according to both the optimization purpose and the evaluation function value. Besides self-learning, every individual in each niche population would imitate and learn from the external benchmarking and the internal benchmarking, and eventually, beyond the benchmarking and grow up into a brand new benchmarking, namely, becoming as the learning object of the other individuals. Therefore, BLA proposed here is a kind of learning-competition optimization model, and also a competition-learning optimization model.

In the process of searching and learning, self-organization learning in each niche population goes like this: each individual will conduct external benchmarking learning first, that is to say, it will adjust its search direction and search step according to the best individual within the whole ecological system. Namely, it will decrease the distance with the external benchmarking. If its evaluation function value cannot be improved, the individual will pass into the internal benchmarking learning stage. It will adjust its search direction and search step according to the best individual within the niche population to which it belongs, namely, it will reduce the distance with the internal benchmarking. If its evaluation function value is not improved yet, the individual will carry out self-learning. It means that the individual will get its dual individual via dual operation. In addition, the best individual in each niche population will be fixed as the program process goes, because every niche population will exchange its best individual with other niche populations. So the internal benchmarking in each population, namely, the learning object of the staff inside each population will be replaced by other internal benchmarking. The three learning operations mentioned above will not be executed according to the order but executed selectively, only when its evaluation function value is not improved through one learning operation, the individual will conduct another one.

The idea of benchmarking learning for optimization problems is distinctive, so it is very important to know how to use the usual encoding methods-float-point encoding method and binary encoding method, to give expression to this idea. In BLA described here, the three learning operations-external benchmarking learning, internal benchmarking learning and self-learning，are extremely helpful to express this idea. The details about the three learning operations can be given as follows.

A. External Benchmarking Learning

Let X_E^{best} be the best individual, whose evaluation function value is maximal or minimal according to the optimization purpose, in the whole ecological system, that is, the global optimal individual and the external benchmarking, let G_E^{best} be its corresponding gene expression, let G_K^i be the gene expression of X_K^i, which is the i-st individual within niche population P_K, then, the external learning rate of X_K^i can be given as Eq. (1)

$$\begin{cases} \max f(x): \quad Grate_K^i = Grate' + f_K^i \big/ \tilde{f}_K - 1 \\ \min f(x): \quad Grate_K^i = Grate' + \tilde{f}_K \big/ f_K^i - 1 \end{cases} \tag{1}$$

Wherein, $Grate'$ stands for the initial value of the external learning rate, f_K^i stands for the value of the evaluation function of X_K^i, $\tilde{f}_K$ stands for the average value of P_K.

If binary encoding method was put to use, external benchmarking learning was carried out by X_K^i, which means that the gene-bits in G_K^i, which are different from that in G_E^{best}, would be replaced by the gene-bits in G_E^{best} with a probability of $Grate_K^i$. That is to say, X_K^i took the initiative to narrow the Hamming distance with X_E^{best}.

If float-point encoding method was involved, external benchmarking learning was conducted by X_K^i, which means that with a probability of $Grate_K^i$, G_K^i would be updated according to Eq. (2) as below. That is to say, X_K^i took the initiative to reduce the Euclidean distance with X_E^{best}.

$$G_K^i = G_K^i + \lambda(G_E^{best} - G_K^i) \tag{2}$$

Wherein, $\lambda \in [0,1]$, stands for the shift step length of individual X_K^i. Experiments show that the optimization effect will be better if λ is proportional to the search space, or fixed dynamically according to the evaluation function value in the process of learning. But this is not the focal point of this paper, so it will not be taken into further discussion.

B. Internal Benchmarking Learning

Let X_K^{best} be the best individual, whose evaluation function value is maximal or minimal according to the optimization purpose, in niche population P_K, namely, the local optimal individual and the internal benchmarking; let G_K^{best} be its corresponding gene expression; let G_K^i be the gene expression of X_K^i, which is the i-st individual in niche population P_K. Then, the internal learning rate of X_K^i can be given as Eq. (3).

$$\begin{cases} Binary: \quad Brate_K^i = Brate' - HD_{k,h} \big/ Length + 1 \\ Float: \quad Brate_K^i = Brate' - ED_{k,h} \big/ Radius + 1 \end{cases} \tag{3}$$

Wherein, $Brate'$ stands for the initial value of the internal learning rate; $HD_{K,h}$ stands for the Hamming distance between X_K^i and X_K^{best}; $Length$ stands for the length of the gene expression encoding; $ED_{K,h}$ stands for the Euclidean distance between X_K^i and X_K^{best}, namely, $ED_{k,h} = \sqrt{\sum_1^n (x_i^{best} - x_i)^2}$. $Radius$ stands for the diameter of the search space, that is, $Radius = \sqrt{\sum_1^n (b_i - a_i)^2}$. Here, x_i is the i-st dimension of the gene expression, and $x_i \in [a_i, b_i]$.

Similar to external benchmarking learning, if binary encoding method was put to use, internal benchmarking learning was carried out by X_K^i, which means that the gene-bits in G_K^i, which are different from that in G_K^{best}, would be replaced by the gene-bits in G_K^{best} with a probability of $Brate_K^i$. That is to say, X_K^i took the initiative to narrow the Hamming distance with X_K^{best}.

If float-point encoding method was adopted, internal benchmarking learning was carried out by X_K^i, which means that with a probability of $Brate_K^i$, G_K^i would be updated according to Eq. (4) as below. That is to say, X_K^i took the initiative to diminish the Euclidean distance with X_K^{best}.

$$G_K^i = G_K^i + \lambda(G_K^{best} - G_K^i) \quad (4)$$

Wherein, $\lambda \in [0,1]$, stands for the shift step length of individual X_K^i.

It seems that there is no difference between external and internal benchmarking learning, because they are all apt to narrow the Hamming (or Euclidean) distance between an individual and the best one. As a matter of fact, there is a great difference between external and internal benchmarking learning. It is of importance to reduce the Hamming (or Euclidean) distance. For one thing, it is helpful for the populations to carry out intensive search, which contributes to forming the cluster effect and seeking out the global optimal solution quickly. For another, it is very helpful to maintain the population diversity, because the learning object of each individual within the population is changing constantly and dynamically, therefore, the clustering hierarchy in the whole ecological system is changing dynamically as well.(please refer to **Section 4**: Population Diversity in BLA)

C. Self-Learning

Let X_K^i be the i-st individual in niche population P_K; let f_K^i be the evaluation function value of X_K^i; let $\tilde{f}_K$ be the average value of the niche population P_K; let G_K^i be the gene expression of X_K^i. Then, the self-learning rate of X_K^i can be given as below.

$$\begin{cases} \max f(x): & Srate_K^i = Srate' * \tilde{f}_K / f_K^i \\ \min f(x): & Srate_K^i = Srate' * f_K^i / \tilde{f}_K \end{cases} \quad (5)$$

Wherein, $Srate'$ stands for the initial value of the self-learning rate;

If binary encoding method was put to use, self-learning was carried out by X_K^i, which means that each gene-bit in G_K^i will conduct dual mapping [10] with a probability of $Srate_K^i$ shown as below (Fig. 1).

primary gene:	1	0	1	0	0	1	0	1	1	0	0	0	1	0	0
dual gene:	0	1	0	1	1	0	1	0	0	1	1	1	0	1	1

Fig. 1 Binary dual mapping

If float-point encoding method was put to use, X_K^i carried out self-learning means that X_K^i would make use of logistic chaos mapping to help itself to jump out of the current region. Let $G_K^i = [x_1, x_2, \cdots, x_{n-1}, x_n]$, $x_i \in [a_i, b_i]$. Then G_K^i would be updated according to Eq. (6) as below

$$\begin{cases} \lambda_i(0) = \dfrac{x_i - a_i}{b_i - a_i} \\ \lambda_i(t+1) = \delta\lambda_i(t)(1-\lambda_i(t)) \\ x_i(t) = a_i + \lambda_i(t)(b_i - a_i) \\ \delta \in [2,4], i = 1,2,3,\cdots,n \end{cases} \quad (6)$$

From Eq. (5), it is easy to see that when the optimization purpose is to obtain the maximum value of the evaluation function, if the evaluation function value of some individual is smaller than the average value of the niche population it belongs, its self-learning desire will be heightened quickly and its self-learning rate will increase to a larger number, and so it is more likely to obtain its dual individual to enhance the evaluation function value. But if its evaluation function value is bigger than the average value of the niche population it belongs, its self-learning desire will fade away quickly and its self-learning rate will decrease to a smaller number, and so it is great helpful to protect against the ascendant genes from being destroyed. In like manner, when the optimization purpose is to obtain the minimum value of the evaluation function, the self-learning desire will adjust accordingly.

D. Pseudo Code

Let $E = \{P_1, P_2 \ldots P_{np}\}$ be the whole ecological system consisting of np niche populations, N_i be the number of individuals in P_i, P_i^j be the jth individual in P_i, P_i^{best} be the best individual in P_i. Let f_i^j be the evaluation function value of P_i^j, $\tilde{f}_i$ be the average value of P_i at current generation, $\tilde{f}_E$ be the average value of E at current generation, P_{best} be the best individual in E. Let $Grate'$ be the external learning rate, $Brate'$ be the internal learning rate,

$Strate'$ be the self-learning rate. Let $\max\_gen$ be the maximum iteration times. Then the pseudo code for BLA can be given as below.

(1) Initialize the np, N, $Great'$, $Brate'$, $Srate'$ and other parameters if necessary.

(2) *for gen=1:max_gen, do*

(a) *for i=1:np, do*

i . Evaluate f_i^j

ii . Evaluate $\tilde{f}_i$

iii. Find and record p_i^{best}

(b) Find out and record P_{best}

(c) Find out, record and update the best individual so far in E.

(d) evaluate $\tilde{f}_E = \left(\sum \tilde{f}_i\right) / np$

(e) for i=1:np, do

i . P_i^j conduct external benchmarking learning

ii . if f_i^j does not be improved, then, P_i^j will conduct internal benchmarking learning

iii. if f_i^j does not be improved yet, then, P_i^j will carry out self-learning

(f) *if* $\tilde{f}_E$ does not be improved or the best individual in E does not be replaced *do*

p_i will exchange its best individual with other niche populations.

(3) Output the global optimal solution.

III. MAIN FEATURES AND ADVANTAGES

The main features and advantages of BLA, which are different from and superior to other optimization methods, were described, analyzed and given as below.

A. The Unique Search Model

The general framework of evolutionary algorithms (EAs) represented by genetic algorithm (GA) is that all the individuals in ecological system are carrying out genetic operations including selection, crossover and mutation at random. Though the crossover rate and mutation rate can be fixed adaptively to reduce the randomness of the genetic operation, EAs appeared more stochastic and less intelligent. In general, the search strategy of EAs is still passively adaptive. In the particle swarm optimization (PSO), particles are apt to follow both the best position they had drifted and the position of local optimal particle, so the search behavior of particles within PSO is intelligent in a certain extent. However, PSO cannot maintain the population diversity natively and cannot get over the premature convergence problem. What is more, no other encoding methods, but only the float-point encoding method can be used in PSO. A kind of PSO based on binary encoding method was proposed [2, 37] by Kennedy and Eberhart, but it ran at the sacrifice of PSO’s limited intelligence and became purely a kind of random search algorithm. The artificial fish swarm algorithm (AFSA) was designed through simulating the behavior patterns of fish swarm, such as prey, swarm and follow. Like PSO, AFSA is not able to keep the population diversity and overcome the premature convergence problem either. The ants in the ant colony optimization (ACO) would find out the best route according to the principle - the thicker the pheromones, the closer the route. ACO takes advantage of positive feedback mechanism, but it cannot get over the distraction of the local extremum. Besides that, ACO is suitable for discrete problem like Traveling Salesperson Problem (TSP), and it is hard to convergence to the global optimal solution when the number of cities is too great. The simulated annealing (SA) makes use of the Metropolis acceptance criteria to help to jump out of the local extremum regions, but it includes too many repetitive iterations and this requirement is difficult to be met in the practical applications. Therefore, SA is just a heuristic random process based on Monte Carlo method. The taboo search (TS) and the predatory search (PS) are more likely to act as a unique search strategy and search pattern. The former would flag the region which had been searched to reduce the repetitive iterations, and the later has no detailed computing method, it is just a strategy to balance the global search and local search, and only a way to keep the algorithm better both in exploration and exploitation.

The general framework of BLA proposed in this paper is that all the individuals in ecological system are selectively executing learn-actions by themselves and their learning objects are constantly changing. The purpose and direction of the learn-actions are very definite and clear, namely, every individual in ecological system wants to grow up and become the learning object of other individuals by simulating and learning from the best individual. When some niche population find out a new global or local optimal solution, that is, setting a new external or internal benchmarking, it will attract lots of individuals, who belong to other niche populations, to join in and help to search a better global or local optimal solution. But if the niche population cannot seek out a new global or local optimal solution all the time, its members will be all attracted into other niche populations until its extinction. In a similar way, if some individual can chase down a better global optimal solution, it will attract a lot of individuals into its local region to create and form a new niche population. This is the dynamic niche technique proposed in this paper. Therefore, BLA is not only a learning-competition optimization model, but a competition-learning optimization model. All in all, BLA described here, whose framework is brief and clear, is easy to be programed. It is a learning-search strategy as well as a competitive optimizing method. Obviously, BLA, a new kind of search model designed based on management theory and method in business term, is different from all existing optimization methods designed based on the principles of the natural world, especially the biological world.

B. Less Repetitive Operations in Search Process

One of the remarkable characteristics of BLA is less repetitive operations in the search process. A great many of repetitive genetic manipulations would be conducted in the process of evolutionary algorithms (EAs) represented by genetic algorithms (GAs). For example, an individual may carry out crossover operation with one more individuals in the search process, yet its corresponding objective function value did not return to the program immediately. And due to the impact of random selection, after cross-operation was completed, some individuals within the population may be not involved in the crossover operation and their gene structures did not change, while some individuals within the population may conduct cross-operation for many times and their good gene structure may have been damaged. In the standard genetic algorithm (SGA) and the vast majority of its improved versions, each individual's objective function value was evaluated after the three core genetic operations, namely selection, crossover and mutation, was carried out in sequence, which in fact is equal to that each individual conducted a series of repetitive genetic manipulations before its objective function value was evaluated. Additionally, the individual in simulated annealing (SA) will conduct a large number of exploratory movements in the thermal equilibrium phase, which makes no contribution to find out the global optimal solution in most of the time. Though the particles in particle swarm optimization(PSO) have no repetitive movements in the search process, if the objective function value was not improved after once drifting, only at the next iteration were the search direction and drift step of particles corrected. In the search process of BLA, individual's learning behavior is conducted selectively, namely on condition that its objective function value was not be improved after carrying out previous learning strategy, the individual would conduct the next learning strategy, which not only contributes to create no useless and repetitive actions, but also helps to amend the gene structure before once search process came to an end.

C. No Useless Operations in Search Process

Useless operations, here, refers in particular to some operations in algorithm, which make no contribution to the optimization problems in question. Take the function optimization problems with real type of decision variables as an example, if the float-point encoding method was adopted, then a great deal of useless operations would appear in the search process of the most of popular optimization methods, such as PSO, SA and AFSA. The search operations in these methods often make the individuals fly out of solution space, namely the decision variables are apt to go beyond the scope of the variable range and become illegal solutions after a series of search operations. To solve this problem, the general approach is that if some decision variable is over the boundary, it would be replaced by the boundary value or modulus of itself, or it would be assigned a new value within the variable range. But as a matter of fact, these solutions are not only another form of repetitive operations, in essence, but a useless operation, which completely destroyed the intelligence of the methods and was not conducive to solving the problems in question. In the search process of BLA, every individual in the niche population is carrying out benchmarking-oriented search actions which would narrow the Euclidean distance with the benchmarking. Because the learning object is legitimate, the individual would not fly out of the solution space. Therefore, there are no useless operations in BLA as PSO, SA and AFSA did, this will be verified by the function optimization experiments in the later section.

IV. POPULATION DIVERSITY IN BLA

A. Theory Analysis

When dealing with dynamic problems, any slight changes in the environmental variables or/and constraints might stir up significant volatility in the external environments of population, and sometimes even cause deformation of search space, such as changes in the dimension, so the optimal solution for the problem was no longer fixed. Therefore, the top priority of the methods designed for these dynamic problems is detecting the smooth or acute changes in the environments at any time and reacting to them as quick as possible. Besides that, it is also very important to track the trajectory of the optimal solution in the search space. Based on this demand, how to maintain the population diversity in search process is of the most importance for any optimization algorithms. The traditional static-oriented optimization methods tend to find out the global optimal solution via the ultimate convergence of population, so they are inevitably prone to lose the population diversity prematurely in the search process. As opposed to the traditional static-oriented optimization methods, BLA proposed here is not in pursuit of the ultimate convergence of population, but keeping the population diversity throughout the search process, so it can adapt to solving dynamic problems. In order to obtain the global optimal solution, a bulletin board was set to record the global optimal solution in the current generation and the records would be updated after the each iteration. Finally, the latest record in the bulletin board is exactly the global optimal solution.

Traditionally, the population convergence was defined according to the distance among the individuals within population (when binary encoding was involved, the distance refers to Hamming distance; when float-point encoding adopted, the distance refers to Euclidean distance). Population convergence means that the distance between any two individuals within the population tend to zero in some probability, that is to say, the population convergence was achieved when all the individuals within the population had the same gene expression. Therefore, maintaining the population diversity means that plenty of individuals, among which the distance is non-zero, are kept within the population. In the search process of BLA proposed in this paper, each individual's learning behavior is essentially that each individual adjusted its gene expression in terms of benchmarking, namely learning object. Each individual within the ecological system was likely to reduce the distance with the global best individual after external benchmarking learning and narrow the distance with the local best individual after internal benchmarking learning. However, with the appearance of the new external benchmarking and the exchange of the internal benchmarking, each individual's learning object was changing as well, so it is not probable that all the individuals within the ecological system would have the same genotype. What is more, every individual was more likely to add its distance with others after self-learning, because the self-learning strategy in BLA, especially the logistic chaos mapping, would help each individual to jump out of the current region including local extremum. In fact, the function of self-learning is equivalent to re-initialization, so each individual would arrive to any area within the search space after self-learning. Therefore, from this point of view, BLA is non-convergence naturally, which ensured the population diversity within the whole ecological system and each niche population, and ensured the exploration of BLA in the search process as well. This had been confirmed at the next section for experimental analysis.

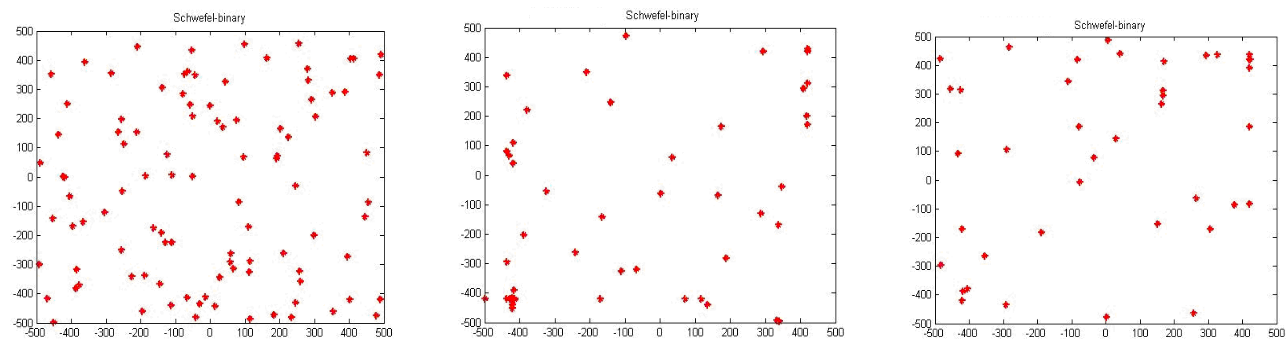
B. Experimental Analysis

The function shown in Eq. (7) is a famous testing function known as Schwefel's function, it was used to test BLA's ability to keep population diversity in the search process.

$$\min f(x) = \sum_{i=1}^{2} [-x_i \sin(\sqrt{|x_i|})] \quad x_i \in [-500, 500] \tag{7}$$

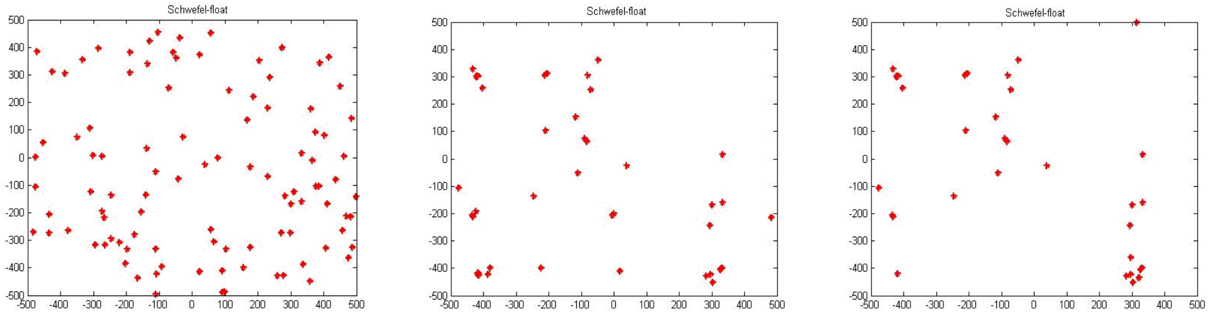
There were 10 niche populations in the ecological system and each niche population contained 10 individuals, so there were 100 individuals within the whole ecological system. The initial value of external learning rate, internal learning rate and self-learning rate were all set as 0.5. If binary encoding method was put to use, each individual's gene code length was set as 50, yet if float-point encoding method was adopted, the shift step length of each individual was set as 0.618. The maximum iteration times was set as 100.

Now the maps of population distribution in three periods - prime, middle and final, were shown as below.



(2-1) prime (2-2) middle (2-3) final

Fig. 2 Maps of population distribution when binary encoding was put to use



(3-1) prime (3-2) middle (3-3) final

Fig. 3 Maps of population distribution when float-point encoding was put to use

From Fig. 2 and Fig. 3 shown as above, we can see that the diversity of population had been always kept in the search process of BLA. In each periods of the search process, although the global or local best individuals brought about cluster effect in some small scopes, it did not lead to damage to the population diversity of the entire ecological system. This is in consistent with the theory analysis and forecast previously described. As a matter of fact, if the parameters of BLA were set appropriately, the performance of BLA would be more excellent.

V. EXPERIMENT AND SIMULATION

In order to test the static search capability and the dynamic tracking ability, BLA would compare with several well-known methods in a static function optimization problem and a dynamic function optimization problem shown as follows.

A. Static Function Optimization Problem

Static function optimization problem means that the domain of function, individual encoding and other factors remain the same in the optimizing process, the optimal solution of the function in question would be sought out according to the optimization purpose. There are some test functions shown in Table 1 as below.

TABLE 1 TEST FUNCTIONS USED

Name	*Function*	*Domain*	*OptimalPosition*	*OptimalSolution*
Sphere	$f_1(x)=\sum_{i=1}^{D} x_i^2$	$[-100,100]^D$	0.0^D	0
Griewank	$f_2(x)=\frac{1}{4}\sum_{i=1}^{D} x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	$[-600,600]^D$	0.0^D	0
Rastrigin	$f_3(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$	$[-5.12,5.12]^D$	0.0^D	0
Rosenbrock	$f_4(x)=\sum_{i=1}^{D-1}(100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2)$	$[-30,30]^D$	1.0^D	0
Ackley	$f_5(x)=-20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}}-e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)}+e+20$	$[-32,32]^D$	0.0^D	0
Schwefel	$f_6(x)=418.9829D-\sum_{i=1}^{D}(x_i\sin(\sqrt{\lvert x_i\rvert}))$	$[340,500]^D$	420.9687^D	0
Step	$f_7(x)=\sum_{i=1}^{D}(\lvert x_i+0.5\rvert)^2$	$[-100,100]^D$	-0.5^D	0
Schwefel'sP 2.22	$f_8(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	$[-10,10]^D$	0.0^D	0
Quadric	$f_9(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2$	$[-100,100]^D$	0.0^D	0
QuadricNoise	$f_{10}(x)=\sum_{i=1}^{D}\lvert ix_i^4\rvert+rand[0,1)$	$[-1.28,1.28]^D$	0.0^D	0

To compare with BLA proposed in this paper, some famous optimization methods including standard genetic algorithm [1]. (SGA), simulated annealing [2] (SA), particle swarm optimization [6] (PSO), artificial fish swarm algorithm [38] (AFSA), have been put to use in this experiment. And float-point encoding method was adopted in all these methods. Although all the controls parameters involved in these methods are very different, to ensure a fair comparison as possible, the total number of individuals in all these methods were all set as 100 except SA, and the maximum iteration times of these methods were all set as 10,000. Besides that, other crucial controls parameters involved were set as below.

SGA: tournament selection strategy, arithmetic crossover and Gaussian mutation was adopted, crossover rate and mutation rate were set as 0.7 and 0.2 respectively. SA: the iteration number was set as 20 for each phase of thermal equilibrium to compensate for the disadvantages of single individual's serial search; the spherical neighborhood was constructed and its radius was set as 20; the number of iterations was set as the current annealing temperature. PSO: the full model was used to update the particle drift velocity; the inertia weight would decrease from 0.9 to 0.1 linearly; the two acceleration factors were all set as 2; the maximum speed was set as the variation range of each variable; ring topology neighborhood would be constructed on the basis of index number. AFSA: the perceived distance of artificial fish was set as 20; the number of attempts to prey was set as 5; the shift step length was set as 0.3; the crowding factor was set as 0.618; the behavior of artificial fish was chose due to the principle of progress. BLA: at the initialization stage, 10 niche populations and 10 individuals in each niche population, the initial value of external learning rate and internal learning rate were all set as 0.5, and the initial value of self-learning rate was set as 1; the shift step length of each individual was set as 0.382.

To test the performance of each method and its stability, four indexes, which included the number of times for successfully finding out the global optimal solution (NG), the mean value of the global optimal solution (MV), the standard deviation (SD), were used to compare each method's optimization effect for the ten test functions shown as above. All the data in Table 2 reflect the average value of 50 experimental results.

TABLE 2 TEST RESULT WHEN THE D WAS SET AS 50

		F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**
SGA	**NG**	11	8	21	10	9	8	6	26	7	23
	MV	4.35E+0	1.44E+0	9.22E+1	2.87E+3	8.77E+0	3.63E-2	8.20E+0	4.35E+0	4.38E+3	2.83E-2
	SD	6.35E+3	5.22E-2	5.41E+1	4.37E+3	9.24E+3	3.55E-1	5.46E+3	6.35E+0	2.50E+3	7.80E-3
SA	**NG**	11	9	17	9	17	9	11	17	10	17
	MV	8.61E-1	3.19E-1	3.49E+0	1.44E+1	5.49E-1	7.43E-2	9.07E-1	8.99E-1	8.44E+0	1.88E-3
	SD	1.40E+1	8.61E-2	6.80E+2	3.29E+2	7.22E+0	6.04E-1	9.37E-2	9.06E-2	5.39E+1	8.94E-2
PSO	**NG**	18	16	9	15	27	16	31	20	18	33
	MV	6.55E-1	3.82E-2	3.66E-2	6.07E-2	8.04E-2	4.25E-4	7.44E-2	7.61E-2	6.01E-1	2.70E-5
	SD	2.08E+0	6.07E-1	2.29E+1	7.64E+1	7.34E+0	3.57E-2	5.06E-1	4.42E-2	3.55E+0	5.01E-2
AFSA	**NG**	17	21	20	18	15	28	19	21	14	20
	MV	8.04E-2	6.88E-1	9.34E-3	7.37E-3	3.55E-2	6.90E-5	7.99E-3	6.07E-3	8.59E-2	3.72E-5
	SD	5.89E-1	2.83E-2	8.08E+0	9.97E+0	7.57E+0	7.50E-1	7.33E+0	3.24E-2	4.66E-1	8.88E-2
BLA	**NG**	46	44	48	46	43	47	47	45	45	44
	MV	5.27E-8	4.07E-8	5.54E-9	5.37E-8	3.61E-7	4.04E-8	2.90E-8	3.72E-7	8.66E-6	8.34E-8
	SD	5.33E-5	8.32E-5	6.60E-7	2.04E-6	3.55E-8	6.37E-5	6.33E-7	8.55E-6	3.73E-5	4.05E-9

Hardware Platform: CPU: AMD Athlon (tm) 64 X2 Dual Core Processor 3600+, 1.91GHz;DDR:667MHz,1024MB

Software Platform: Matlab7.6

From Table 2, it is easy to see that in 50 experiments, BLA successfully found out the global optimal solution of Function **F1** for 46 times, yet SGA, SA, PSO, and AFSA were just 11 times, 16 times, 18 times and 17 times, respectively; for Function **F2**, BLA successfully sought out its global optimal solution for 44 times, yet SGA, SA, PSO, and AFSA were just 8 times, 9 times, 16 times and 21 times, respectively; for Function **F3**, SGA, SA, PSO, and AFSA still had a poor performance. In terms of indicator NG, MBV and SD, the overall performance of AFSA was close to PSO. On the whole, SA is inferior to other four methods, partly because SA is a kind of optimization method based on single individual's serial search, unlike other four methods, which are based on the parallel search of population. Additionally, the iterations of thermal equilibrium stage in these experiments were not big enough. Consequently, the annealing process was not fulfilled adequately. Similarly, for the other test functions, BLA performed very well. The test showed that BLA is far superior to other methods.

B. Dynamic Function Optimization Problem

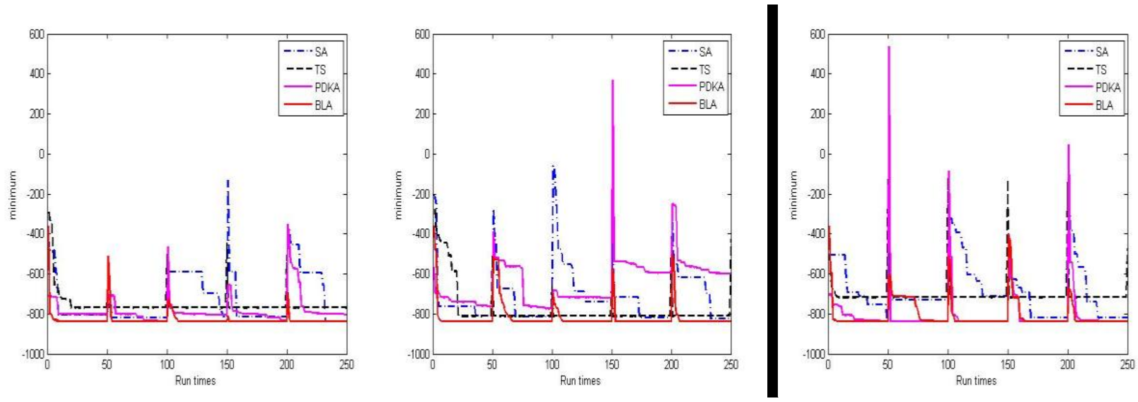
It is true that some scientific literatures about structuring dynamic function are still rarely seen, Angeline proposed a kind of mobile parabolic problem [39], which involved in a single hump function based on float-point encoding. Brank and Mattfeld etc. put up a method structuring dynamic function [40]. Namely, the goal is to find out the maximum of all the peaks, whose position, height and width would change with slight changes in the environments. It is easy to construct all kind of mobile peak functions in this way, yet its process is rather trivial.

In the static function optimization problem, float-point encoding was adopted by all the optimization algorithms, so binary encoding would be put to use in this experiment. In terms of binary encoding involved here, the dynamic environments can be constructed with the same method as the dynamic bit matching problem. Namely, a binary string, template called here, was defined, whose length is equal to the gene expression of an individual. Then all of the individuals would carry out bit XOR operation with the template. The dynamic function optimization here means that the pre-given template would change dynamically in the running course of algorithm, so each individual's gene expression would change accordingly. As a result, if some bit in the template is 0, then the corresponding bit in individual's gene expression will remain unchanged, however, if the bit in the template is 1, then the corresponding bit in individual's gene expression will turn into its opposite value. Therefore, the ratio between the number of bits whose value is 1 and the length of the template, can be used to characterize the intensity of environmental changes.

To compare with BLA, several well-known optimization methods including simulated annealing (SA), taboo search [41] (TS), primal dual genetic algorithm [42] (PDGA), have been put to use to optimize Schwefel's function shown in Eq. (7). In this experiment, binary encoding was adopted in all the optimization methods and the length of individual's gene expression was set as 50, namely, a control variable was represented by 25 binary bits. The purpose of this experiment is to test each method's ability to track the extreme point of Schwefel's function in the dynamic environments. The controls parameters in these methods were set as below.

SA: the iteration number was set as 80 for each phase of thermal equilibrium to compensate for the disadvantages of single individual's serial search; each individual's neighbor solutions were created by turning some bit in the gene expression into its opposite value; the number of iterations was set as the current annealing temperature. TS: each iteration, 10 neighbor candidate solutions were created in the same way as SA; Tabu list consisted of the optimal solution after each iteration. PDGA: tournament selection strategy, anti-XOR crossover and single-point mutation were adopted; crossover rate and mutation rate were set as 0.7 and 0.2 respectively. BLA: at the initialization stage, 10 niche populations and 10 individuals in each niche population, the initial value of the external learning rate, internal learning rate and self-learning rate were all set as 0.5.

The maximum iteration times in each method was set as 250, a template would be created every 50 iteration and all of the individuals would conduct bit XOR operation with the template, so environments went through 5 times of oscillation. r stands for the ratio between the number of bits whose value is 1 and the length of the template, namely, the intensity of environments changes. When r is equal to 0.1, 0.5 and 0.9 respectively, each method's ability to track the extreme point of Schwefel's function in the dynamic environments was shown as below.



(4-1) r =0.1 (4-2) r =0.5 (4-3) r =0.9

Fig. 4 Comparison of four methods' dynamic tracking ability

From Fig. 4, we can see that the search performance of SA and TS was unstable, they could not find out the global optimal solution, in other words, they could not track the trajectory of the maximum point after the environments changed, especially the environmental changes were acute. When the environments changed smoothly, as shown in Fig. (4-1) and Fig. (4-2), PDGA could not effectively track the trajectory of the maximum point, yet if the environments changed sharply, as shown in Fig. (4-3), PDGA had a very outstanding performance in tracking the trajectory of the maximum point, this is mainly because its dual-mapping strategy played an important role. The great majority of individuals within the population had greatly deviated from the optimization purposes after the environments changed acutely, but the dual mapping was able to help these individuals bounce back to the high yielding regions. Additionally, the dynamic environments were constructed by dynamic temples, so PDGA performed excellently; in fact, if the dynamic environments were constructed through other methods, for example, changing the scope of decision variables, the dual-mapping strategy would do nothing for PDGA. In the present experiment, the performance of SA was slightly better than TS on the whole, just because at each stage of thermal equilibrium, SA iterate 80 times, which was seven times more than TS. However, neither SA nor TS found out the global optimal solution in the new environments. In contrast, BLA had much more powerful search capabilities and adaptability. It would adjust its search direction and search step length to adapt to the new environments after detecting the smooth or acute changes of the environments, so BLA could search and find out the global optimal solution as soon as possible. From Fig. 4, it is easy to find that there are a bit of slight bending in these value curves, this demonstrated that BLA is extremely sensitive to the changes of the environments and it could work friendly with the environments through transparent interaction, which could help BLA seek out a better solution than the current one and improve it continuously. It is also easy to see that BLA was always able to seek out the global optimal solution at the beginning of each new cycle, and no matter how the intensity of environmental changes was, these global optimal solutions made no difference, this indicated that the search performance of BLA is very stable.

It must be explained that the maximum iteration times of each method was set as 250, there are 5 periods, and each period included only 50 times iteration. Therefore, for a relatively large search space is concerned, the search capabilities of SA and TS could not come into play adequately. Nevertheless, BLA was able to search and find out the global optimal solution in the specified period. It makes clear that the search strategy of active learning, which is indeed able to interact with the environments friendly and help to discover and look out the better feasible solution, is much better than the strategy of random search and passive adaptive search.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a competitive learning mechanism based on dynamic niches was set up according to the core values of benchmarking, in which the active learning-based search strategy took the place of the traditional passive adaptive search strategy. Consequently, some defects of the existing intelligent optimization methods (EIOMs), for example, the running direction of the search process was indecisive, the EIOMs could not maintain the population diversity and could not be adapt to dynamic optimization problems, were all got overcome. As a result of the imitation and learning to the benchmark, individuals within the population are able to approach to the target regions in the solution space and seek out the optimal solutions quickly. The search behaviors of these individuals are no longer apt to be completely passive, self-adaptive and random, but active and direction-oriented. What is more, the formidable problem of maintaining the diversity of population was completely overcame through the self-organizing learning process of the niche system and its friendly interaction with the environments, thus, the exploration and exploitation of the BLA will be balanced self-adaptively. And then, BAL is able to accurately detect the slight changes of the environments and track the trajectory of the extreme points in the search space, and thus, BLA is naturally adaptable for the dynamic environments.

BLA, originated from benchmarking theory of business management, is different from the EIOMs, which stem from the biological activities of nature. Therefore BLA is brand new and it is a newborn member of the family comprising the modern intelligent optimization methods. However, similar to other optimization methods, BLA also involved a number of controls parameters, such as learning rates, etc. and how to set these controls parameters to optimize BLA to achieve the best effect, which itself is also a combinatorial optimization problem, is one of our next research topics.

REFERENCES

[1] J. H. Holland, “Adaptation in Natural and Artificial Systems,” University of Michigan Press, Arbor, 1975.

[2] S. Kirkpatrick, C. D. Gelatt, and Jr, M. P. Vecchi, “Optimization by simulated Annealing,” *Science*, vol. 220, pp. 671-680, 1983.

[3] M. Creutz, “Microcanonical Monte Carlo Simulation,” *Physical Review Letters*, vol. 50, iss. 19, pp. 1411-1414, 1983.

[4] P. Moscato, “On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Pasadena,” California, USA: Tech. Rep. Caltech Concurrent Computation Program, Report 826, Callifornia Institute of Technology, 1989.

[5] A. Colorni, M. Dorigo, and V. Maniezzo, “Distributed optimization by ant colonies,” in *Proceedings of the first European conference on artificial life*, 1991, vol. 142, pp. 134-142.

[6] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of IEEE international conference on neural networks*, 1995, vol. 4, no. 2, pp. 1942-1948.

[7] R. Storn and K. Price, “Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.

[8] L. N. de Castro and F. J. von Zuben, Artificial Immune Systems: Part I-Basic and Applications, vol. 1, pp. 76-85, 1999.

[9] L. N. de Castro and F. J. von Zuben, “The Clonal Selection Algorithm with Engineering Applications,” in *Workshop Proceedings of GECCO, Workshop on Artificial Immune Systems and Their Applications*, Las Vegas, USA, July 2000, pp. 36-37.

[10] M. M Eusuff and K. E.Lansey, “Shuffled frog leaping algorithm: a memetic meta-heuristic for combinatorial optimization,” *J. Heuristics*, in press, 2000.

[11] H. Murase, “Finite element inverse analysis using a photosynthetic algorithm,” *Computers and Electronics in Agriculture*, vol. 29, no. 1, pp. 115-123, 2000.

[12] I. Zelinka and J. Lampinen, “SOMA - self-organizing migrating algorithm,” In *6th International Conference on Soft Computing*, 2000.

[13] Z. W. Geem, J. -H. Kim, and G. V. Loganathan, “A new heuristic optimization algorithm: harmony search,” *Simulation*, vol. 76, no. 2, pp. 60-68, 2001.

[14] K. M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” *IEEE Control Systems Magazine*, vol. 22, pp. 52-67, 2002.

[15] X. F. Xie, W. J. Zhang, and Z. L. Yang, “Social Cognitive Optimization for Nonlinear Programming Problems,” in *Proceedings of the First International Conference on Machine Learning and Cybernetics*, Beijing, 2002.

[16] R. Wood and A. Bandura, “Social cognitive theory of organizational management”, *Academy of management Review*, vol. 14, no. 3, pp. 361-384, 1989.

[17] X. L. Li, *A New Intelligent Optimization Method-Artificial Fish School Algorithm*, Zhejiang University, 2003.

[18] Y. H. Zhou and Z. Y. Mao, “A New Search Algorithm for Global Optimization: Population Migration Algorithm,” *Journal of South China University of Technology (Natural Science)*, vol. 31, no. 3, 2003.

[19] S. Nakrani and C. Tovey, “On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers,” *Adaptive Behavior-Animals, Animats, Software Agent, Robots, Adaptive Systems*, vol. 12, no. 3-4, pp. 223-240, 2004.

[20] D. Karaboga and B. Basturk, “On the performance of artifical bee colony Algorithm,” *Applied Soft Computing*, vol. 8, no. 1, pp. 687-697, 2008.

[21] X. S. Yang, “New enzyme algorithm,Tikhonov regulation and inverse parabolic analysis, in Advances in Computational Methods in Science and Engineering,” Lecture Series on Computer and Computer Sciences, ICCMSE 2005, Eds. T. Simons and G. Maroulis, vol. 4, pp. 1880-1883.

[22] K. N. Krishnanand and D. Ghose, “Detection of multiple source locations using a glowworm metaphor with application to collective robotics,” in *Proceedings of IEEE Swarm Intelligence Symposium*, 2005, pp. 84-91.

[23] L. Tong, C.-F. Wang, W.-B. Wang, and W.-L. Su, “A Global Optimization Bionics Algorithm for Solving Integer Programming - Plant Growth Simulation Algorithm,” *Systems Engineering-theory & Practice*, vol. 01, pp. 76-85, 2005.

[24] A. Mucherino and O. Seref, “Monkey Search: A Novel Meta-Heuristic Search for Global Optimization,” *AIP Conference Proceedings 953, Data Mining, System Analysis and Optimization in Biomedicine*, 2007, pp. 162-173.

[25] T. C. Havens, C. J. Spain, N. G. Salmon, and J. M. Keller, “Roach Infestation Optimization,” *Swarm Intelligence Symposium*, 2008, IEEE, pp. 1-7.

[26] X. S. Yang, “Firefly algorithm” in Nature-inspired Metaheuristic Algorithms, chapter 8, Luniver Press, 2008.

[27] D. Simon, “Biogeography-based optimization,” *IEEEE Transactions on Evolutionary Computation*, vol. 6, no. 12, pp. 702-713, 2008.

[28] Y. Zhang, “Research on Optimization Searching Method of Optical Principle,” Harbin Engineering University, 2009.

[29] H.-P. Ma, Z.-D. Chen, and Z.-X. Pan, “Evolutionary algorithm based on species migration optimization, Control and Decision,” vol. 24, no. 01, pp. 1620-1624, 2009.

[30] A. Tero, S. Takagi, and T. Saigusa, et a1., “Rules for biologically inspired adaptive network design,” *Science*, vol. 327, pp. 439-442, 2010.

[31] L. P. Xie and J. C. Zeng, “Physicomimetics Method for Global Optimization,” *Journal of Computer Research and Development*, vol. 48, no. 5, pp. 848-854, 2011.

[32] C. Ju and H. Xue-qing, “Explosion Search Algorithm with Neighborhood Search Mechanism,” Computer Engineering, vol. 37, no. 18, pp. 183-187, 2011.

[33] T. Shi-heng and Y. Wei-yu, “New algorithm for global optimization:cell membrane optimization,” *Application Research of Computers*, vol. 28, no. 12, pp. 454-457, 2011.

[34] W. T. Pan, “A new evolutionary computation approach: Fruit Fly Optimization Algorithm,” *2011 Conference of Digital Technology and Innovation Management*, Taipei.

[35] W. T. Pan, “A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example,” *Knowledge-Based Systems*, vol. 26, Feb. 2012.

[36] Y. Dong-long, *Benchmarking: How to Learn from Benchmark Enterprises*, Beijing, China: China Social Sciences Press, 2004 (in Chinese).

[37] J. Kennedy and R. C. Eberhart, “A discrete binary of the particle swarm optimization,” in *Proc. IEEE International Conference on systems, man and cybernetic*, 1997, Piscataway, NJ: IEEE Service Center, pp. 4104-4209.

[38] L. Xiao-lei, S. Zi-jiang, and Q. Ji-xin, “An optimizing method based on autonomous animals: fish-swarm algorithm,” *System Engineering Theory and Practice*, vol. 22, no. 11, pp. 32-38, 2002.

[39] P. J. Angeline, “Tracking extrema in dynamic environments,” in *Proceedings of the 6th international conference on Evolutionary Programming*, 1997, vol. 1213, pp. 335-345.

[40] J. Brank, D. Mattfeld, and T. Engelmann, “A simple dynamic scheduling benchmark,” *Evolutionary Optimization in Dynamic Environments*, Kluwer Academic Publishers, 2002.

[41] F. Glover, Tabu search-Part Ⅰ, *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190-206, 1989.

[42] S. Yang, “Non-stationary problem optimization using the primal-dual genetic algorithm,” Evolutionary Computation, CEC’03, The 2003 Congress on IEEE, vol. 3, pp. 2246-2253, 2003.