[Downloaded from www.aece.ro on Saturday, October 13, 2018 at 09:27:49 (UTC) by 159.226.100.198. Redistribution subject to AECE license or copyright.]

Particle Swarm Optimization with Power-Law Parameter Based on the Cross-Border Reset Mechanism

Haoyun WANG[1,2,3], Yuhan FEI[1], Yibai LI[1], Shougang REN[1,3], Jianhua CHE[1,3], Huanliang XU[1,2,3,*]
[1]*College of Information Science and Technology, Nanjing Agricultural University, Nanjing 210095, China*
[2]*Agricultural Engineering Postdoctoral Research Station, Nanjing Agricultural University, Nanjing 210031, China*
[3]*Jiangsu Collaborative Innovation Center of Meat Production and Processing, Quality and Safety Control, Nanjing 210095, China*
wanghy@njau.edu.cn, 2016201004@njau.edu.cn, 2016114003@njau.edu.cn, rensg@njau.edu.cn chejianhua@njau.edu.cn, huanliangxu@njau.edu.cn

***Abstract*—In order to improve the performance of traditional particle swarm optimization, this paper introduces the principle of Levy flight and cross-border reset mechanism. In the proposed particle swarm optimization, the dynamic variation of parameters meets the power-law distribution and the pattern of particles transition conforms to the Lévy flight in the process of algorithm optimization. It means the particles make long distance movements in the search space with a small probability and make short distance movements with a large probability. Therefore, the particles can jump out of local optimum more easily and coordinate the global search and local search of particle swarm optimization. This paper also designs the cross-border reset mechanism to make particles regain optimization ability when stranding on the border of search space after a long distance movement. The simulation results demonstrate the proposed algorithms are easier to jump out of local optimum and have higher accuracy when compared with the existing similar algorithms based on benchmark test functions and handwriting character recognition system.**

***Index Terms*—evolutionary computation, optimization, particle swarm optimization, performance evaluation, benchmark testing.**

I. INTRODUCTION

With the rapid development of human society, people have to face much more complicated questions and have higher requirement for the optimal solution. Therefore, the outstanding optimization algorithms become more important. In the early 1990s, scholars proposed swarm intelligence optimization algorithms, which were inspired by the social activities mechanism of animals and insects in the nature [1]. Particle swarm optimization (PSO), which belongs to swarm intelligence optimization, has been widely used in particular. The mathematical operations of PSO are simple and pervasive because only a few parameters need to be adjusted. Moreover, it doesn't need higher requirement for the performance of CPU and RAM. The distributed parallel algorithms for PSO enhance the capability of processing large quantity of data and improve the execution speed. In recent years, there are many problems need to be solved about PSO. For instance, how to jump out of local optimum and improve search accuracy, how to reduce computational complexity and enhance convergence speed. These problems need to be solved especially when dealing with complex problems [2].

The algorithm can jump out of local optimum by balancing global and local search ability [3]. According to related researches, we learned the best search patterns for particular targets are the explosive, intermittent and occasional ones, which conform to the power-law distribution, rather than the distinct, systematic and regular ones [4]. Therefore, we introduce the principle of Lévy flight to improve the traditional PSO, in which the value of parameters follow the power-law distribution and the pattern of particles movement conformed to Lévy flight [5]. The particles make long distance movement in the search space with small probability and make short distance movement with large probability, which makes the particles can jump out of local optimum more easily. It provides new insights to solve the disadvantage of traditional PSO. This paper also proposed the cross-border reset mechanism to make the particles regain optimization ability when they stranded on the border of search space after a long distance movement. The proposed algorithms are compared with existing similar algorithms based on benchmark test functions and the handwriting character recognition system which developed by our group.

II. RELATED WORK

The swarm intelligence optimization has been increasingly used in the fields of engineering and economics. Scholars have proposed a series of bionic swarm intelligence algorithms, which typically include artificial ant colony algorithm, PSO, artificial fish algorithm, artificial bee colony algorithm, shuffled frog leaping algorithm, and

This work was supported in part by the National Natural Science Foundation of China (No. 31601545); the Fundamental Research Funds for the Central Universities (No.KJQN201732); The Agricultural Project for New Variety, New Technology and New Model of Jiangsu Province (No.SXGC[2014]309); Government Sponsored Project Under the Science and Technology Development Initiative : Research and Demonstration of Electronic Food Safety Traceability (No. 2015BAK36B05); Key Research and Development Program of Jiangsu Province, Research and Demonstration on Key Technologies of Food Safety Risk Early Warning and Traceability Integrated Service Platform (No. BE2016803).

Haoyun WANG and Yuhan FEI equally contributed to this work.

Digital Object Identifier 10.4316/AECE.2017.04008

1582-7445 © 2017 AECE

[Downloaded from www.aece.ro on Saturday, October 13, 2018 at 09:27:49 (UTC) by 159.226.100.198. Redistribution subject to AECE license or copyright.]

firefly algorithm, etc [6-7]. Kar [8] showed that PSO is effective, easy to operate and can be widely used in many fields. However, PSO lack an effective mechanism to escape of local optimum and has other problems worthy to research [9]. In recent years, the researchers improve the performance of PSO by several methods, including adjusted the parameters [10], optimized the topology structure [11], used improved hybrid optimization algorithm and introduced biological mechanism [12]. This paper adjusts the parameters of algorithm by Lévy flight dynamically and uses the cross-border reset mechanism to improve the performance of algorithm.

The values of parameters w, c_1, c_2 has an important role in the search process. The original PSO don't have inertia weight, which make the global search ability and local search ability unbalanced. Shi et al. [13] introduced the concept of inertia weight into the original PSO and proposed the standard particle swarm optimization (SPSO) in 1998. To make up for the shortcoming of linear decreasing inertia weight, Shi et al. [14] put forward fuzzy rules to adjust inertia weight according to the features of test functions. Clerc [15] proposed a random adjustment for inertia weight and Chatterjee [16] put forward a way of nonlinear inertia variation for dynamic adjustment in PSO. About the settings of c_1 and c_2, the traditional PSO uses fixed learning factors to achieve the best balance between global search and local search ability. For different problems, the ranges of c_1 and c_2 are both 1.0 to 2.5 in general. Suganthan [17] considered the best values of c_1 and c_2 are invariable. However, Ratnaweera et al. [18] used a linear function to adjust the learning factors, which make c_1 lessen and c_2 largen gradually and Zhang put forward a self-adjusting strategy of c_1 and c_2 based on the fitness values of the particles.

To improve the performance of algorithm, researchers begin to introduce biological mechanism into PSO and related algorithms appeared constantly, which have been proved effective in practical applications. Liu [19] used the flight mechanism of geese migration to improve the PSO's performance. Inspired by the symbiotic coevolution between species in nature, Chen [20] proposed a multi-species PSO, which extends the dynamics of the canonical PSO by taking into account species extinction and speciation events. Qin [21] was inspired by biological parasitic behavior and proposed a two species PSO, which refers to facultative parasitic behavior between hosts and parasites. Yang [22] proposed a new PSO based on the operator of chemotaxis in the bacterial foraging, which is easy to search for the optimal value in region.

In 1996, Viswanathan et al. [23] proposed the biological mechanism named as Lévy flight by establishing the link between animal foraging behavior and random walk theory for the first time. They used GPS to research albatross foraging behavior and found the flight ranges of albatross following the power-law distribution, which consist with the foresight of Shlesinger a decade ago [24]. Lévy flight is speculated the most effective foraging pattern when the foods are scattered over a large area. Nowadays, Lévy flight is widely used to the optimization algorithm, which makes search efficiency maximization under uncertain environment. Literature [25] introduced the principle of Lévy flight into the PSO and proposed the several improved PSO which based on Lévy flight, including the algorithm's step transfer obey the power-law distribution (Levy Bare Bones), the algorithm which based on the hyperspheres (Levy Pivot) and the part of parameters obey the power-law distribution (Levy PSO). It's worth noting that Levy PSO will be abbreviated as LPSO in this paper. These several algorithms lay a solid foundation to the further improvement about PSO which based on the Lévy flight and provide new insights into the improvement of PSO. Wang [26] proposed an earthquake disaster emergency rescue model based on cooperation mechanism and Lévy flight to reduce the blindness and randomness of the rescue work. Li [27] proposed a variant of cooperative quantum-behaved PSO with two mechanisms to reduce the search space and avoid the stagnation, which are dynamic varying search area and Lévy flight mechanism. Yan [28] put forward an improved bacterial foraging optimization algorithm based on Lévy flight and Xie proposed an improved bat algorithm based on Lévy flights and differential operators. In 2014, Hakli [29] proposed a novel particle swarm optimization algorithm with Lévy flight (LFPSO). In the proposed method, a limit value is defined for each particle, and if the particles could not improve self-solutions at the end of current iteration, the limit is increased. If the limit value determined is exceeded by a particle, the particle is redistributed in the search space with Lévy flight method. Experimental results show that the LFPSO is more successful than well-known and recent population-based optimization methods.

In this paper, we introduce the power-law distribution into the dynamic variation of parameters (w, c_1, c_2) and make the step transfer follow the power-law distribution to enhance the ability of particles to jump out the local optimum. Furthermore, the coefficient of Lévy flight was no longer use the experience values which rely on the large number of according to the problems appeared in the experiments, we propose the cross-border reset mechanism to improve the convergence accuracy of the algorithm and enhance the ability of particle to jump out of local optimum. Finally, the performance and accuracy of the proposed algorithms will be examined on well-known benchmark functions, comparing with SPSO and LFPSO as well. Furthermore, we have found researchers always test the performance of algorithms by practical application in the further step, Castillo O, et al. [30] test the performance of improved ant colony optimization by fuzzy control of a mobile robot, Martín D, et al. [31] test the performance of multi-objective genetic algorithm by a complex electromechanical process, Harmanani H M, et al. [32] test the performance of improved genetic algorithm by open-shop scheduling problem and Precup R E, et al. [33] test the nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models by anti-lock braking systems. In this paper, we will apply the proposed algorithms into handwriting character recognition system which developed by our group to test the performance of these algorithms as well.

III. THE PSO BASED ON THE LÉVY FLIGHT AND CROSS-BORDER RESET MECHANISM

This paper adjusts the value of parameters (w, c_1, c_2) follow the power-law, which makes the pattern of particles

[Downloaded from www.aece.ro on Saturday, October 13, 2018 at 09:27:49 (UTC) by 159.226.100.198. Redistribution subject to AECE license or copyright.]

transition conform to Lévy flight. During the simulation, we found when a long distance movement happened, the particles have high possibility to strand on the border of search space and lost optimization ability. To solve this problem, we design the cross-border reset mechanism and propose particle swarm optimization with power-law parameter based on the cross-border reset mechanism (PLP-PSO-CBR). Referring to the framework of LFPSO [29], we embed the cross-border reset mechanism into LFPSO and proposed LFPSO-CBR. The two improved algorithms which based on cross-border reset mechanism will be described in this chapter and the performance will be analyzed in later chapters.

3.1 The Basic Principle of PSO

The principle of PSO can be described as follow. It is assumed that the search space is M-dimensional and the number of particles is *N*. The position of the *i*th particle at time t is expressed as $\mathbf{X}_i(t)=(x_i^1(t),x_i^2(t),...,x_i^M(t))$, of which the historical optimal position is expressed as $\mathbf{P}_{best_i}=(p_i^1,p_i^2,...,p_i^M)$ according to the fitness of the *i*th particle. The optimal value among $\mathbf{P}_{best_{i,i=1,2,3,...,N}}$ is recorded as $\mathbf{G}_{best}=(G^1,G^2,...,G^M)$ according to the fitness of all particles. The velocity of particles transition at time *t*+1 is defined as $\mathbf{V}_i(t+1)=(v_i^1(t+1),v_i^2(t+1),...v_i^M(t+1))$ and the position of the *i*th particle at time *t*+1 is replaced according to the following equations.

$$V_i^d(t+1)=w\cdot v_i^d(t)+c_1\cdot rand\cdot(p_i^d-x_i^d(t))+ \\ c_2\cdot rand\cdot(G^d-x_i^d(t)),1\le i\le N,1\le d\le M \tag{1}$$

$$x_i^d(t+1)=x_i^d(t)+v_i^d(t+1),1\le i\le N,1\le d\le M \tag{2}$$

In Eq. (1), the constant c_1 and c_2 are learning factors, w is inertia weight, and rand is a random number between 0 and 1. The ranges of position and velocity in the *d*th dimension are $[-x_{\max}^d,x_{\max}^d]$ and $[-v_{\max}^d,v_{\max}^d]$. PSO will set the position as the boundary value when the particle stranded on the border of the *d*th dimension. The initial position and velocity of particle swarm are generated randomly, and they will be updated according to Eq. (1) and Eq. (2) until the stop condition is satisfied.

3.2 The Principle of Lévy flight

The search pattern of Lévy flight is different from the ordinary pattern because of the randomness. Lévy flight is a type of random walk, of which the step sizes obey power-law distribution and the search directions obey uniform distribution. The proposed algorithms use the generator of Lévy values with Mantegna rule [34]. In Mantegna rule, the step size is designed as follow:

$$s(\beta)=u/|v|^{1/\beta} \tag{3}$$

In Eq. (3), u and v obey normal distribution, $u\sim N(0,\sigma_u^2),v\sim N(0,\sigma_v^2)$. σ_u and σ_v are defined as follows:

$$\sigma_u=1 \tag{4}$$

$$\sigma_v=\left\{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}}\right\}^{1/\beta} \tag{5}$$

In Eq. (5), Γ is standard Gamma function, $\Gamma(n)=(n-1)!$. In this paper, β is coefficient of Lévy flight and no constant value is taken for the β parameter, but a random value in the (0, 2] interval is taken for each new distribution procedure [29]. If the β value randomly so taken takes small values, it allow the particle perform very long jumps in the search space and prevents constantly being trapped in local minima, if big values are attained, it allow the particle perform short movement in the search space and continues to derive new values around the global optimal. The randomization can be more efficient as the steps obey a Lévy distribution which can be approximated by the power-law. Therefore, the steps consist of many small steps and occasionally large-step or long-distance jump. Thus, Eq. (2) can be restated as follow:

$$x_i^d(t+1)=x_i^d(t)+\alpha\oplus levy(\beta) \tag{6}$$

In Eq. (6), α generally is a random number and $\oplus$ is dot product means entry-wise multiplications. In this paper, α is the step size which should be related to the scales of the problem of interest, α is random number for all dimensions of particles as well. Furthermore, $Levy(\beta)$ can be calculated by Mantegna rule [35] as follow:

$$levy(\beta)\sim 0.01\frac{u}{|v|^{1/\beta}},(0<\beta\le 2) \tag{7}$$

In this paper, the parameters (*w*, c_1, c_2) of PSO are generated according to Mantegna rule and obey the power-law distribution. Therefore, the velocity of particles transition is redefined as follow:

$$LV_i^d(t+1)=levy_w(\beta)\cdot v_i^d(t)+ \\ levy_{c_1}(\beta)\cdot rand\cdot(p_i^d-x_i^d(t))+ \\ levy_{c_2}(\beta)\cdot rand\cdot(G^d-x_i^d(t)) \\ 1\le i\le N,1\le d\le M,0<\beta\le 2 \tag{8}$$

In Eq. (8), $levy_w(\beta)$, $levy_{c_1}(\beta)$ and $levy_{c_2}(\beta)$ are produced respectively according to Eq. (3). In the meantime, Eq. (2) can be redefined as follow:

$$x_i^d(t+1)=x_i^d(t)+LV_i^d(t+1), \\ 1\le i\le N,1\le d\le M \tag{9}$$

3.3 PLP-PSO-CBR

When we make the parameters (*w*, c_1, c_2) follow the distribution of power-law, it means the pattern of particles transition will conform to Lévy flight. During the experiments, we found the particles will strand on the border of search space occasionally after a long distance movement, which are unable to search further. To solve this problem, we introduce the cross-border reset mechanism and propose the PLP-PSO-CBR. The mechanism will initialize these particles which stranded on the border in the search space and make the initialized particles regain the optimization ability. The flow chart of PLP-PSO-CBR is shown in Figure 1. In the meantime, we can view the Pseudocode of PLP-PSO-CBR in Table I.

[Downloaded from www.aece.ro on Saturday, October 13, 2018 at 09:27:49 (UTC) by 159.226.100.198. Redistribution subject to AECE license or copyright.]

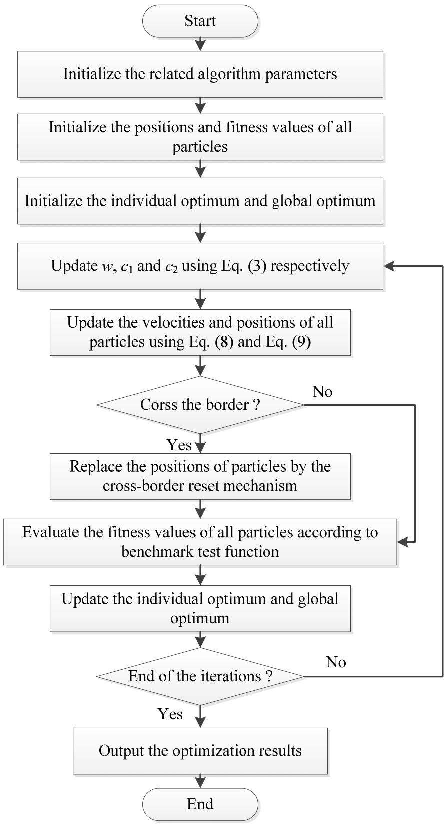

Figure 1. The flow chart of PLP-PSO-CBR

TABLE I. PSEUDOCODE OF PLP-PSO-CBR

Initialize the dimension of search space M, the number of particles N

Initialize the current iteration t, the maximum iterations T

Initialize the position of all particles $\mathbf{X}_{i,i=1,2,3,\ldots,N}(t)$ randomly

Initialize the fitness values of all particles according to benchmark test function

Set the values of individual optimum $\mathbf{P}_{best_{i,i=1,2,3,\ldots,N}}$ and global optimum $\mathbf{G}_{best}$

While t<T do

Start the generator of levy value to update w, c_1, c_2 using Eq. (3)

For i=1:N

For d=1:M

Update the velocity $LV_i^d(t+1)$ of the ith particle using Eq. (8)

Update the position $x_i^d(t)$ of the ith particle using Eq. (9)

Use the cross-border reset mechanism according to whether stranded on the border

End for

Evaluate the fitness value for new position $\mathbf{X}_i(t)$ according to benchmark test functions

If $\mathbf{X}_i(t)$ is better than $\mathbf{P}_{best_i}$

Set $\mathbf{X}_i(t)$ to be $\mathbf{P}_{best_i}$

End if

If $\mathbf{X}_i(t)$ is better than $\mathbf{G}_{best}$

Set $\mathbf{X}_i(t)$ to be $\mathbf{G}_{best}$

End if

End for

t = t + 1

End while

3.4 LFPSO-CBR

In the literature [29], the trial value and limit value are set for each particle. If the trial value is less than the limit value, the particles will move randomly in a small area. Otherwise, the particles make Lévy flight in a large range. Thus, the transition model of the particles can switch between random walk with large probability and Lévy flight with small probability. We also found inappropriate setting of limit value will disturb the balance of global and local search by extensive use of Lévy flight step transfer mode. Referring to the framework of algorithm in literature [29], the LFPSO-CBR is proposed based on the cross-border reset mechanism and aim to prove the validity of this mechanism. The flow chart of LFPSO-CBR is shown in Figure 2.

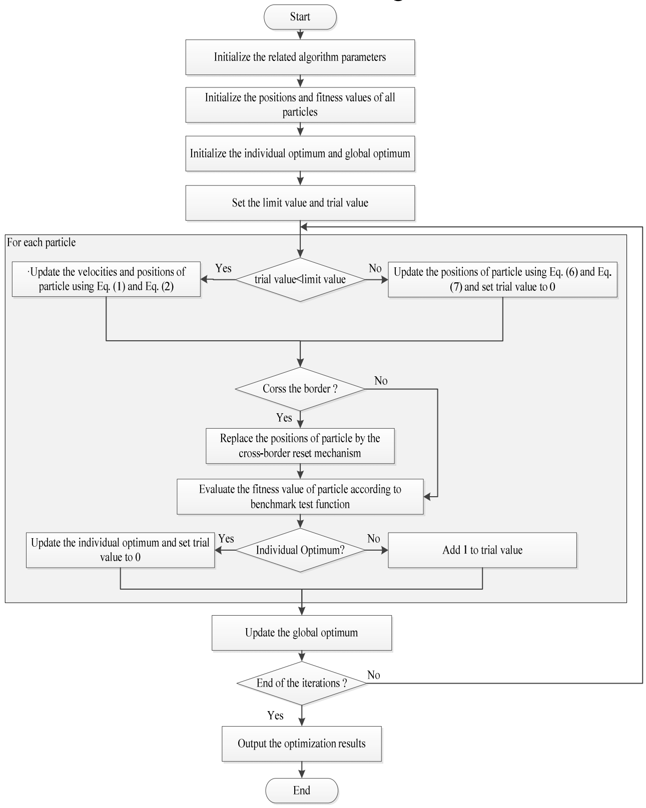

Figure 2. The flow chart of LFPSO-CBR

IV. THE EXPERIMENTAL DESIGN

The experiment is to verify the effectiveness of cross-border reset mechanism and the principle of Lévy flight. We want to prove the algorithms can make the particles jumps out of the local optimum more easily and enhance the search accuracy when we introduce these two methods. LFPSO has shown the superior performance when compared with CLPSO, HPSO-TVAC, FIPSO, SPSO-40, DMS-PSO and other popular swarm intelligence algorithms in recent years, which include group search algorithm (GSO), cuckoo algorithm (CS) and firefly algorithm (FA) [33]. In this paper, the proposed algorithms are compared with LFPSO and SPSO for different types of benchmark test functions. In addition, we will analyze the setting of parameters for the proposed algorithm.

There are 18 kinds of benchmark test functions in this

[Downloaded from www.aece.ro on Saturday, October 13, 2018 at 09:27:49 (UTC) by 159.226.100.198. Redistribution subject to AECE license or copyright.]

experiment, which are mainly derived from the literatures [29], [36-37]. The benchmark test functions can be divided into three types, including unimodal function (U), normal multimodal function (M) and rotated multimodal function (R). The hardware environment for the experiments is Intel(R) Pentium(R) CPU G620 @ 2.60GHz, memory 8.00GB. The software environment is Windows 10 and MATLAB 2012a. The parameter setting about proposed algorithms are referenced in the literature [29], and the specific information is shown in Table II. The benchmark test functions are shown in Table III.

In addition, the algorithms can be freely used via http://123.57.158.232:88/PSO/ and we will update and maintain this website regularly. Researchers can validate and improve these algorithms by this website.

TABLE II. PARAMETER SETTINGS FOR THE ALGORITHMS

Algorithm	SPSO	LFPSO/ LFPSO-CBR	LPSO	PLP -PSO-CBR
Population	50	50	50	50
Dimension	30	30	30	30
Iteration	50000	50000	50000	50000
Inertia weight	0.7213	linear decreasing $\frac{Max\_iter - iter}{Max\_iter}$	None	Power-law distribution $\beta \in (0,2]$
Learning factor	c_1=c_2=1.1931	c_1=c_2=2	Experience Value	Power-law distribution $\beta \in (0,2]$
Limit value	-	5	-	-
Repetition	15	15	15	15

TABLE III. LIST OF BENCHMARK TEST FUNCTIONS

No	Name	Type	Formula
1	Sphere	U	$F_1 = \sum_{i=1}^{n} x_i^2$
2	Step	U	$F_2 = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$
3	Rosenbrock	U	$F_3 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
4	Quartic	U	$F_4 = \sum_{i=1}^{n} i x_i^4$
5	Shifted Sphere	U	$F_5 = \sum_{i=1}^{D} z_i^2 + f\_bias_1, z = x - o, x = [x_1, x_2, ..., x_D]$ $x \in [-100, 100]^D, D: dimmensions.$ $o = [o_1, o_2, ...o_D]: the\_shifted\_global\_optimum.$ $global\_optimum: x^* = o, F_1(x^*) = f\_bias_1 = -450$
6	Shifted Schwefel's Problem 1.2	U	$F_6 = \sum_{i=1}^{D} (\sum_{j=1}^{i} z_j)^2 + f\_bias_2, z = x - o, x = [x_1, x_2, ..., x_D]$ $x \in [-100, 100]^D D: dimmensions.$ $o = [o_1, o_2, ...o_D]: the\_shifted\_global\_optimum.$ $global\_optimum: x^* = o, F_2(x^*) = f\_bias_2 = -450$
7	Rastrigin	M	$F_7 = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$
8	Ackley	M	$F_8 = -20\exp\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\} - \exp\{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\} + 20 + e$
9	Griewank	M	$F_9 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$
10	Schwefel 2.26	M	$F_{10} = 418.98288727243369 \cdot n - \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$
11	Alpine	M	$F_{11} = \sum_{i=1}^{n} \lvert x_i \cdot \sin(x_i) + 0.1 \cdot x_i \rvert$
12	Levy	M	$F_{12} = \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + \lvert x_n - 1 \rvert [1 + \sin^2(3\pi x_n)]$
13	Rotated hyper-ellipsoid	R	$F_{13} = \sum_{i=1}^{N} (\sum_{j=1}^{i} x)^2$
14	Rotated Schwefel	R	$F_{14} = 418.9828 \cdot n - \sum_{i=1}^{n} z_i$ $z_i = \begin{cases} y_i \sin(\sqrt{\lvert y_i \rvert}), if \lvert y_i \rvert \le 500 \\ 0, otherwise \end{cases}, y_i = y'_i + 420.96$ $y' = M \cdot (x - 420.96), M: orthogonal\_matrix$
15	Rotated Rastrigin	R	$F_{15} = \sum_{i=1}^{n} [y_i^2 - 10\cos(2\pi y_i) + 10]$ $y = M * x, M: orthogonal\_matrix$
16	Rotated Ackley	R	$F_{16} = -20\exp\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} y_i^2}\} - \exp\{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi y_i)\} + 20 + e$ $y = M * x, M: orthogonal\_matrix$
17	Rotated Griewank	R	$F_{17} = \frac{1}{4000}\sum_{i=1}^{n} y_i^2 - \prod_{i=1}^{n}\cos(\frac{y_i}{\sqrt{i}}) + 1$ $y = M * x, M: orthogonal\_matrix$
18	Rotated Rosenbrock	R	$F_{18}(x) = \sum_{i=1}^{n-1} [100(y_{i+1} - y_i^2)^2 + (y_i - 1)^2]$ $y = M * x, M: orthogonal\_matrix$

V. RESULTS AND ANALYSIS

In this paper, the algorithms based on the cross-border reset mechanism and Lévy flight are compared with SPSO, LFPSO and LevyPSO (LPSO) to verify the search precision and the ability to jump out of the local optimum. The results are analyzed by MATLAB curve fitting toolbox in the further step. Furthermore, we provide setting of parameters by these results to improve the efficiency of the algorithms.

5.1 The Principle of Experiments

In the experiments, we have got the search space and optimum value about these benchmark test functions previously. The algorithm will find the optimum value in the search space based on different benchmark test functions and the result will getting closer to the optimum value according to fitness function during the optimization. In the experiments, the fitness function is the benchmark test function itself. Fox example, if we choose a benchmark test function to test the performance of algorithm, we need to get the lower-bound and upper-bound of this benchmark test function first and use them to construct the search space. Then, the algorithm will find the optimum value in the search space according to the fitness function. Finally, we can get the optimal solution, error value and convergence of this algorithm. However, the initialization of PSO is stochastic, we need to repeat the algorithm and get the mean value of results to reduce the error of randomness. In addition, the algorithms which include SPSO, LPSO, LFPSO, LFPSO-CBR and PLP-PSO-CBR can be freely used via http://123.57.158.232:88/PSO/ respectively. In the meantime, the benchmark test functions which are mentioned above can be freely used via

[Downloaded from www.aece.ro on Saturday, October 13, 2018 at 09:27:49 (UTC) by 159.226.100.198. Redistribution subject to AECE license or copyright.]

http://123.57.158.232:88/PSO/5.SPSO/benchmark_func.m and http://123.57.158.232:88/PSO/5.SPSO/func.m. Furthermore, the optimum value and search space of each function can be freely used via http://123.57.158.232:88/PSO/5.SPSO/get_fun_info.m.

5.2 Results of Contrast Experiments

In this work, we compare the proposed algorithms which based on the cross-border reset mechanism and Lévy flight with SPSO, LFPSO, LevyPSO (LPSO) for 18 kinds of benchmark test functions and Table IV shows the simulation result. The simulation results contain the mean error and standard deviation error. The rank of algorithm depends on its mean error first and standard deviation error second. When the mean error between algorithms is close to each other, the algorithm with lower standard deviation error has a higher rank. The algorithms are evaluated between level 1 to level 5. The performance of algorithm is judged by the mean rank for the 18 benchmark test functions and we can compare the algorithms by the final rank.

TABLE IV. COMPARISON RESULTS OF ALGORITHMS

No.	Error	SPSO	LFPSO	LPSO	LFPSO-CBR	PLP-PSO-CBR
1	Mean	3.61e+03	1.42e-01	4.28e-01	1.26e-01	**0.93e-01**
	Std.	1.20e+03	2.63e-02	3.61e-01	6.58e-02	9.21e-02
	Rank	5	3	4	2	1
2	Mean	3.83e+03	2.63e+00	5.57e+02	2.80e+00	**1.62e+00**
	Std.	1.37e+03	3.48e+00	3.62e+02	4.33e+00	2.58e+00
	Rank	5	2	4	3	1
3	Mean	1.35e+02	2.97e+01	3.95e+01	**1.23e-02**	2.74e+01
	Std.	6.07e+01	5.48e-01	1.50e+01	1.03e-02	5.32e-01
	Rank	5	3	4	1	2
4	Mean	9.35e-01	4.72e-03	3.36e-02	4.21e-03	**2.16e-03**
	Std.	4.47e-01	2.83e-03	2.45e-02	3.58e-03	2.04e-03
	Rank	5	3	4	2	1
5	Mean	2.03e+04	3.31e+01	5.23e+03	**2.35e+01**	2.90e+01
	Std.	5.93e+03	2.91e+01	4.01e+03	1.60e+01	2.69e+01
	Rank	5	3	4	1	2
6	Mean	3.86e+04	2.03e+03	1.37e+04	1.73e+03	**1.21e+03**
	Std.	1.52e+04	5.33e+02	5.21e+03	5.08e+02	5.56e+02
	Rank	5	3	4	2	1
7	Mean	1.26e+02	3.17e+01	1.38e+02	3.61e+01	**2.11e+01**
	Std.	2.32e+01	2.83e+01	3.26e+01	1.73e+01	2.42e+01
	Rank	4	2	5	3	1
8	Mean	1.25e+01	7.01e-01	0.54e+01	7.09e-01	**3.95e-01**
	Std.	1.81e+00	7.23e-01	3.81e+00	8.14e-01	6.58e-01
	Rank	5	2	4	3	1
9	Mean	3.05e+01	3.73e-01	5.37e-01	3.45e-01	**1.27e-01**
	Std.	1.07e+01	7.02e-02	4.77e-01	1.81e-01	7.82e-02
	Rank	5	3	4	2	1
10	Mean	1.91e+02	1.72e+02	1.90e+02	1.55e+02	**0.54e+02**
	Std.	2.63e+01	6.21e+01	3.46e+01	6.66e+01	5.60e+01
	Rank	5	3	4	2	1
11	Mean	1.40e+01	5.76e-01	3.17e+00	5.25e-01	**2.04e-01**
	Std.	3.63e+00	5.47e-01	2.36e+00	7.49e-01	5.01e-01
	Rank	5	3	4	2	1
12	Mean	1.30e+01	1.79e+00	8.97e+00	**1.14e+00**	1.63e+00
	Std.	5.14e+00	1.53e+00	3.83e+00	1.85e+00	1.15e+00
	Rank	5	3	4	1	2
13	Mean	1.00e+04	3.23e+00	4.71e+01	2.79e+00	**1.76e-01**
	Std.	5.14e+03	1.73e+00	6.52e+01	1.98e+00	1.56e+00
	Rank	5	2	4	3	1
14	Mean	8.03e+02	8.42e-02	8.21e+02	**7.63e-02**	8.22e-02
	Std.	1.25e+02	1.68e+02	1.74e+02	1.98e+02	1.69e-01
	Rank	4	3	5	1	2
15	Mean	5.12e+02	4.97e-02	5.42e+02	**4.23e-02**	4.95e-02
	Std.	1.07e+02	8.37e+01	9.52e+01	8.12e+01	1.47e+00
	Rank	4	3	5	1	2
16	Mean	6.91e+02	5.09e-01	5.52e+02	5.14e-01	**4.96e-01**
	Std.	1.73e+02	1.03e+02	1.04e+02	8.37e+01	1.08e+00
	Rank	5	2	4	3	1
17	Mean	1.08e+03	1.27e-03	1.14e+03	1.04e-03	**1.00e-03**
	Std.	4.47e+01	6.82e+01	6.07e+01	8.03e+01	6.79e-03
	Rank	4	3	5	2	1
18	Mean	1.07e+03	1.37e-02	0.74e+03	**1.00e-03**	1.03e-03
	Std.	7.21e+01	5.26e-01	7.49e+01	6.91e-01	5.76e-02
	Rank	5	3	4	1	2
Mean Rank		4.78	2.56	4.22	1.94	**1.33**
Final Rank		5	3	4	2	**1**

In Table IV, some results are bold. It means the best result for benchmark test functions. The final rank indicate PLP-PSO-CBR has the best performance, LFPSO ranked second, LFPSO-CBR ranked third, LPSO ranked fourth and SPSO is the worst. The results are analyzed in detail as follow.

The results between SPSO and LFPSO show that LFPSO has better performance for different types of benchmark test functions, which owe to the transition mode between random walk and Lévy flight. The results also show that several improved PSO which based on the principle of Lévy flight are better than SPSO, because the power-law distribution makes the pattern of particles transfer conform to the Lévy flight and enhance the ability to jump out of local optimum.

During the experiments of LPSO, we found the particles always stranded on the border of search space after a long distance movement because the particles has lost the optimization ability. However, the cross-border reset mechanism can reset the stranded particles in the search space and make the initialized particles regain optimization ability. Therefore, PLP-PSO-CBR and LFPSO-CBR have better performance than LPSO.

By compare the simulation results between LFPSO and LFPSO-CBR. It can be found the performance of LFPSO-CBR is better than LFPSO. The reason is inappropriate setting of limit value disturbs the balance of global search and local search by extensive use of Lévy flight step transfer mode to different type of benchmark test functions in LFPSO. When we embedded the cross-border reset mechanism into LFPSO, it will help particles which stranded on the border regain the optimization ability. Thus, the performance of algorithm can improve by this method.

The simulation results between PLP-PSO-CBR and LFPSO-CBR both present an excellent performance in the experiments. When compared these two algorithms, we can found PLP-PSO-CBR shows better performance for most of benchmark test functions than LFPSO-CBR, because the limit value is difficult to determine in advance and the setting of limit value in LFPSO-CBR always by people's experience when dealing with different benchmark test functions. Although LFPSO can show a superior performance in several benchmark test functions, the setting of limit value will restricts the universality of the algorithm. Instead, PLP-PSO-CBR make the Lévy flight model integrated into dynamic variation of parameters (w, c_1, c_2). Researchers don't need large number of experiment results to determine the value of parameters and it will enhance the randomness of the algorithm. Therefore, PLP-PSO-CBR can show a strong universality and excellent performance for different type of benchmark test functions.

5.2 Analysis of Experiment Results

According the above analysis, PLP-PSO-CBR shows the best performance among the proposed algorithms. In this section, the relationships between error values, convergence point and iteration number for PLP-PSO-CBR are analyzed by MATLAB curve fitting toolbox. The MATLAB curve fitting toolbox can provide the common fitting functions, including exponential function, fourier function, gaussian

[Downloaded from www.aece.ro on Saturday, October 13, 2018 at 09:27:49 (UTC) by 159.226.100.198. Redistribution subject to AECE license or copyright.]

function, interpolant function, polynomial function, power function, etc. The most appropriate fitting function can be found by this toolbox. In addition, the setting of parameters for PLP-PSO-CBR is given in this chapter and these settings will benefit to the efficiency of the algorithm.

1) Correlation analysis of error value and iteration number

The correlation analysis of error value and number of iterations for 9 kinds of benchmark test functions for PLP-PSO-CBR are shown in Figure 3. We use number of iterations as horizontal axis and error value as vertical axis. The curves in Figure 3 are fitted by MATLAB curve fitting toolbox and the related parameters of the fitted curves are shown in Table V.

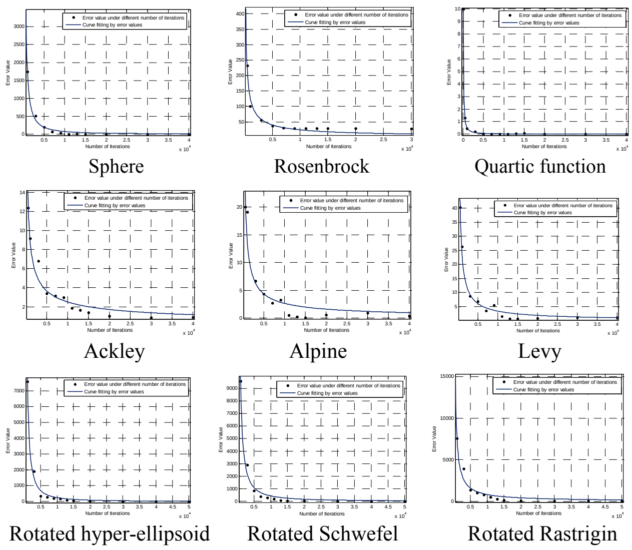

Figure 3. Correlation analysis of error value and number of iterations

TABLE V. RELATED PARAMETERS OF THE FITTED CURVES

Model of fitting function			
$f(x)=a\cdot x^b$			
Benchmark test function	Type	Parameters of function	
		a	b
Sphere	U	2.074e+07	-1.357
Rosenbrock	U	2.081e+04	-1.134
Quartic function	U	3.731e+03	-1.286
Ackley	M	3.922e+02	-0.549
Alpine	M	1.956e+03	-0.717
Levy	M	7.462e+03	-0.835
Rotated hyper-ellipsoid	R	2.154e+08	-1.484
Rotated Schwefel	R	1.069e+08	-1.348
Rotated Rastrigin	R	1.009e+07	-1.237

The results in Figure 3 indicate that the error value show exponential decreasing trends with the increasing numbers of iterations based on different types of the benchmark test functions. It means PLP-PSO-CBR won't fall into local optimum until the iteration number reach to threshold. According to the fitted curves in Figure 3, it can be found the threshold is around 10000 by limiting the angulations of fitted curves greater than 179 degrees when the total iteration number is 50000. Therefore, it is not necessary to execute the algorithm until the final iteration. The iteration can be stopped after 10000 iterations and it can reduce the execution time of PLP-PSO-CBR greatly.

The best model of fitting function for the relation between error value and number of iterations is the power function $f(x)=a\cdot x^b$, which is found by MATLAB curve fitting toolbox. Moreover, the coefficient of power function will change in a certain range for different types of benchmark test functions. The coefficient b ranges between -1.3 to -1.1 when the benchmark test function is unimodal function, the coefficient b ranges between -0.8 to -0.5 when the benchmark test function is multimodal function and the coefficient b ranges between -1.4 to -1.2 when the benchmark test function is composite function. These results demonstrate the convergence speed of the algorithm depends on the types of benchmark test functions. The more complex benchmark test function is, the slower it convergence speed.

2) Correlation analysis of convergence point and total iteration number

The correlation analysis of convergence point and total iteration number for 9 kinds of benchmark test functions for PLP-PSO-CBR are shown in Figure 4. We use total iteration number as horizontal axis and convergence point as vertical axis. If the different between error value of two adjacent iteration numbers less than 1.00e-03, the previous iteration number is recorded as the convergence point. The convergence points of PLP-PSO-CBR for different total iteration number are described in Figure 4. The curves in Figure 4 are fitted by use of MATLAB curve fitting toolbox and the related parameters of the fitted curves are shown in Table VI.

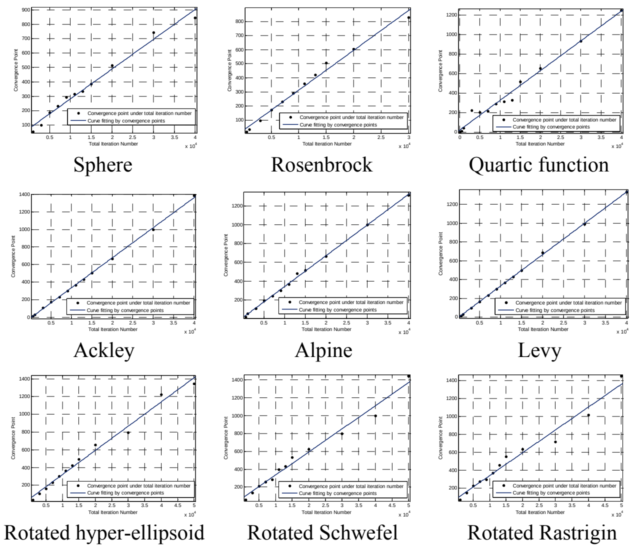

Figure 4. Correlation analysis of convergence point and iteration numbers

TABLE VI. RELATED PARAMETERS OF THE FITTED CURVES

Model of fitting function			
$g(x)=p_1\cdot x+p_2$			
Function	Type	Parameters of function	
		$p1$	$p2$
Sphere	U	0.02074	73.85
Rosenbrock	U	0.02829	29.78
Quartic function	U	0.03043	22.98
Ackley	M	0.03414	-3.96
Alpine	M	0.03266	16.73
Levy	M	0.03329	0.85
Rotated hyper-ellipsoid	R	0.02732	47.03
Rotated Schwefel	R	0.02584	76.08
Rotated Rastrigin	R	0.02553	83.31

The results in Figure 4 indicate that the convergence point show positive correlation with the total iteration numbers. The convergence points become higher with the increase of the total iteration numbers. It means the excessive setting of total iteration numbers won't improve the search precision because error value shows exponential decline with iteration numbers. Thus, the total iteration number needs to be set in

[Downloaded from www.aece.ro on Saturday, October 13, 2018 at 09:27:49 (UTC) by 159.226.100.198. Redistribution subject to AECE license or copyright.]

a reasonable range and the time complexity of the algorithm can be reduced to some extent.

The best model of fitting function for the relationship between convergence point and total iteration numbers is polynomial function $g(x) = p_1 \cdot x + p_2$, which is found by MATLAB curve fitting toolbox. The coefficient of polynomial function fluctuates from 0.02 to 0.03 for different types of benchmark test functions, which indicates the relation between convergence point and total iteration number has less to do with the type of benchmark test functions. Moreover, the total iteration numbers for different type of benchmark test functions can be estimated according to the required precision.

3) The convergence of PLP-PSO-CBR

In this paper, the convergence of this algorithm was analyzed by four coefficients. We use the correlation of error value and iteration number, correlation of convergence point and total iteration number to analyze the convergence by different benchmark test functions. In Figure 3, we can find PLP-PSO-CBR presents good convergence based on different type of benchmark test functions, it can jump out of local optimal more easier and get global optimal regardless of the type of functions. In the meantime, we found the algorithm has already converged before the end of the iteration. Therefore, it is not necessary to execute the algorithm until the end of iteration and we can set the stop point according to the total number of iterations. It will reduce the execution time of algorithm greatly. In Figure 4, we can find the convergence point presents positive correlation with the total iteration numbers based on different types of the benchmark test functions. Thus, the total iteration number needs to be set in a reasonable range and the time complexity of the algorithm can be reduced to some extent. Based on these results, we not only can prove the algorithm has a strong convergence, but also get some empirical values to improve the performance of the algorithm.

VI. Application

In general, we always use convolutional neural network (CNN) to recognize the handwritten images/characters and adjust the weight of CNN by back-propagation neural network (BPN). In this paper, we train the handwriting character recognition system by MNIST database of handwritten digits, this dataset contains 60,000 training sample data and 10,000 test data. Each data size is 28pi * 28 pi. We can use CNN and BPN to realize the recognition function in this system. In CCN, the first layer is the convolution layer, the number of convolutions is 2, the size of the convolution kernel is 5pi * 5pi. The activation function is sigmoid. The second layer is the mean pooling layer and the pool size is 2pi * 2pi. The third layer is the fully connected layer and the connection size is 288pi * 10pi. The activation function is sigmoid. In BPN, we use the batch gradient descent method to test the data, the batch size is 100 and the learning rate is 0.5. Original system is shown in Figure 5.

In addition, the application can be freely used via http://123.57.158.232:88/PSO/Application. Training data and test data which involved in this paper can be freely used via http://123.57.158.232:88/PSO/Application/Data. Furthermore, train_x is the training data that contains 60,000 image data, train_y is the corresponding tag for the training data, test_x is the test data that contains 10,000 image data and test_y is the tag corresponding to the test data.

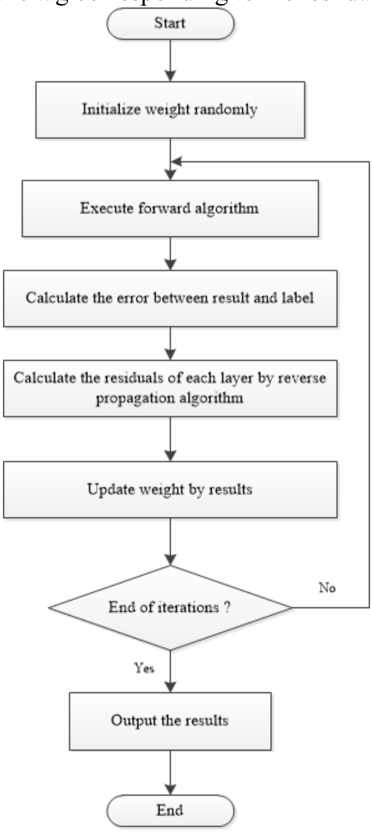

Figure 5. Original system

In this section, we have improved the prediction accuracy of handwriting character recognition system by PLP-PSO-CBR and verified the performance of this algorithm by simulation results. Improved system is shown in Figure 6.

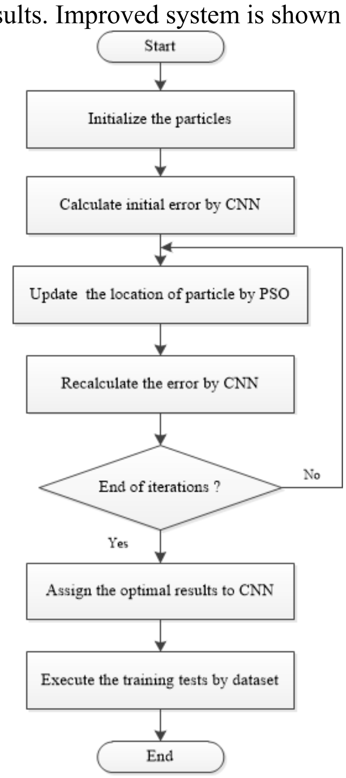

Figure 6. Improved system

In machine learning, there are two indicators measure the performance of the system. One is the fitting ability, the

[Downloaded from www.aece.ro on Saturday, October 13, 2018 at 09:27:49 (UTC) by 159.226.100.198. Redistribution subject to AECE license or copyright.]

other is the generalization ability. The fitting and generalization ability are both important for the system. We need to strike a balance between them to ensure the efficient of the system. The hardware environment for the experiments is Intel(R) Pentium(R) CPU G620 @ 2.60GHz, memory 8.00GB. The software environment is Windows 10 and MATLAB 2012a. The parameters' setting about algorithms is shown in Table VII. The training number or batch number is the CNN training process which uses the batch gradient descent method.

TABLE VII. PARAMETER SETTINGS FOR THE ALGORITHMS

Algorithm	SPSO	LFPSO/ LFPSO-CBR	LPSO	PLP -PSO-CBR
Population	50	50	50	50
Dimension	52	52	52	52
Iteration	500	500	500	500
Inertia weight	0.7213	linear decreasing $\frac{Max\_iter - iter}{Max\_iter}$	None	Power-law distribution $\beta \in (0,2]$
Learning factor	c_1=c_2= 1.1931	c_1=c_2=2	Experience Value	Power-law distribution $\beta \in (0,2]$
Limit value	-	5	-	-
Repetition	10	10	10	10
Training number	600*20	600*20	600*20	600*20
Batch number	12000	12000	12000	12000
Dataset	60000	60000	60000	60000

On the one hands, we can evaluate the fitting ability of the algorithm by Figure 7 - Figure 11. When comparing the PLP-PSO-CBR with other algorithms, we can find 1) It has a fast convergence rate, which ensures the system can find the global optimal in a short time. 2) It can achieve higher convergent accuracy, which leads to higher prediction accuracy in handwriting character recognition system. 3) It has lower initial error, which proves PLP-PSO-CBR is more effective for initial parameter optimization of CNN.

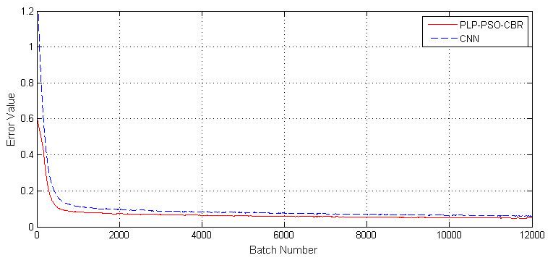

Figure 7. CNN vs PLP-PSO-CBR

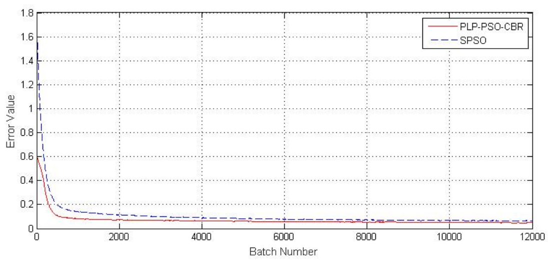

Figure 8. SPSO vs LPSO-CBR

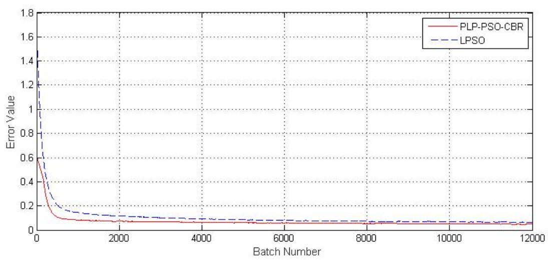

Figure 9. LPSO vs PLP-PSO-CBR

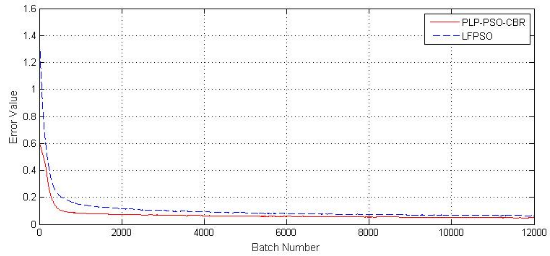

Figure 10. LFPSO vs PLP-PSO-CBR

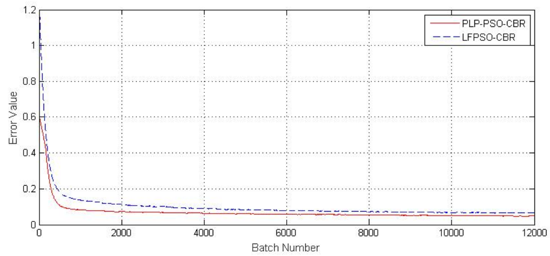

Figure 11. LFPSO-CBR vs PLP-PSO-CBR

TABLE VIII. RELATED PARAMETERS OF THE FITTED CURVES

Algorithms	Error of Prediction	Rank
PLP-PSO-CBR	4.16%	1
LFPSO-CBR	5.92%	3
LFPSO	6.62%	4
LPSO	7.36%	5
SPSO	8.83%	6
CNN	5.21%	2

On the other hands, we can evaluate the generalization ability of this system by Table VIII. We estimated the error of prediction in handwriting character recognition system by different algorithms and found the accuracy of prediction is significantly improved when we use PLP-PSO-CBR to optimize this system. In summary, we can use PLP-PSO-CBR to perfect the handwriting character recognition system and improve the performance of this system significantly by PLP-PSO-CBR when compared with other algorithms.

VII. CONCLUSION

In this paper, we propose the improved PSO with power-law parameter based on the cross-border reset mechanism. In this work, the parameters of algorithm meet the power-law distribution and the pattern of particles transition conforms to the Lévy flight. The cross-border reset mechanism is designed to make the particles regain optimization ability when they stranded on the border. Results demonstrate the method which embedded the cross-border reset mechanism into power-law PSO can enhance the ability to jump out of the local optimum, this method also able to improve the accuracy of algorithm as well. PLP-PSO-CBR presents the best performance among other similar algorithms based on different benchmark test functions. In order to demonstrate the superiority of PLP-PSO-CBR, we apply PLP-PSO-CBR into the handwriting character recognition system. According to the results, we can find the convergence speed of PLP-PSO-CBR is faster, the accuracy of PLP-PSO-CBR is higher, the initial error of system is lower and it can jump out of local optimal more easier when compared with other algorithms in this system.

In addition, we use the MATLAB curve fitting toolbox to analyze the parameters in this algorithm. According to the relationship between error value and iterations, it shows the performance of algorithm will be affected by benchmark test functions and the error value began to stabilization after a

[Downloaded from www.aece.ro on Saturday, October 13, 2018 at 09:27:49 (UTC) by 159.226.100.198. Redistribution subject to AECE license or copyright.]

certain number of iterations. Therefore, we can set the termination of iterations manually and it can reduce the time-consuming of the algorithm to some extent. According to the relationship between convergence point and number of iterations, it shows that the convergence point will move backward by the increase number of iterations, this means we should set the number of iterations in a reasonable range. Furthermore, we found the correlation of convergence point and number of iterations has a weak relationship by different type of benchmark test functions and we can calculate the number of iterations to terminate the algorithm in advance. For solving the higher dimensional optimization problem, cooperative game theory can be introduced to reduce the dimensionality in the future work.

REFERENCES

[1] Yang X S, "Swarm intelligence based algorithms: a critical analysis", Evolutionary Intelligence, vol. 7, no. 1, pp. 17-28, 2014. doi:10.1007/s12065-013-0102-2

[2] Rini D P, Shamsuddin S M, "Particle Swarm Optimization: Technique, System and Challenges", International Journal of Computer Applications, vol.1, no. 1, pp. 33-45, 2011. doi:10.5120/ijais-3651

[3] Saini S, Rambli D R B A, Zakaria N, et al, "A Review on Particle Swarm Optimization Algorithm and Its Variants to Human Motion Tracking", Mathematical Problems in Engineering, vol. 2014, pp. 13-14, 2014. doi:10.1155/2014/704861

[4] Devadoss S, Luckstead J, Danforth D, et al, "The power-law distribution for lower tail cities in India", Physica A Statistical Mechanics & Its Applications, vol. 442, pp. 193-196, 2016. doi:10.1016/j.physa.2015.09.016

[5] Thelwall M, "The discretised lognormal and hooked power-law distributions for complete citation data: Best options for modelling and regression", Journal of Informetrics, vol.10, no.2, pp. 336-346, 2016. doi:10.1016/j.joi.2015.12.007

[6] Yang X S, "Efficiency Analysis of Swarm Intelligence and Randomization Techniques", Journal of Computational & Theoretical Nanoscience, vol. 9, no. 2, pp. 189-198, 2012. doi:10.1166/jctn.2012.2012

[7] Odili J B, Kahar M N M, "Anwar S. African Buffalo Optimization: A Swarm-Intelligence Technique", Procedia Computer Science, vol. 76, pp. 443-448, 2015. doi:10.1016/j.procs.2015.12.291

[8] Arpan Kumar Kar, "Bio inspired computing – A review of algorithms and scope of applications", Expert Systems with Applications, vol. 59, pp. 20-32, no. C, 2016. doi:10.1016/j.eswa.2016.04.018

[9] Li Q, Zhang C, Chen P, "An improved ant colony algorithm based on particle swarm optimization", Control and Decision, vol. 6, pp. 873-878, 2013. doi:10.13195/j.cd.2013.06.75.liq.016

[10] Mandal D, Ghoshal S P, Bhattacharjee A K, "Design of Concentric Circular Antenna Array with Central Element Feeding Using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach and Evolutionary Programing Technique", Journal of Infrared Millimeter & Terahertz Waves, vol. 31, no. 6, pp. 667-680, 2010. doi: 10.1007/s10762-010-9629-9

[11] Jianan Lu, Yonghua Chen, "Particle Swarm Optimization (PSO) Based Topology Optimization of Part Design with Fuzzy Parameter Tuning", Computer-Aided Design and Applications, vol. 11, no. 1, pp. 62-68, 2013. doi: 10.1080/16864360.2013.834139

[12] Mehdinejad M, Mohammadi-Ivatloo B, et al, "Solution of optimal reactive power dispatch of power systems using hybrid particle swarm optimization and imperialist competitive algorithms", International Journal of Electrical Power & Energy Systems, vol. 83, pp. 104-116, 2016. doi:10.1016/j.ijepes.2016.03.039

[13] Shi Y, Eberhart R, "Modified particle swarm optimizer", IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, vol. 6, pp. 69-73, 1998. doi:10.1109/ICEC.1998.699146

[14] Shi Y, Eberhart R, "Fuzzy adaptive particle swarm optimization", Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 101-106, 2001. doi: 10.1109/CEC.2001.934377

[15] Clerc M, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Evolutionary Computation", Proceedings of the 1999 Congress on Evolutionary Computation, vol. 3, pp. 1951-1957, 1999. doi:10.1109/CEC.1999.785513

[16] Chatterjee A, Siarry P, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization", Computers & Operations Research", vol. 33, no. 3, pp. 859-871, 2006. doi: 10.1016/j.cor.2004.08.012

[17] Suganthan P N, "Particle swarm optimiser with neighbourhood operator", Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on IEEE, vol. 3, pp. 1958-1962, 1999. doi:10.1109/CEC.1999.785514

[18] Ratnaweera A, Halgamuge S K, Watson H C, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 240-255, 2004. doi:10.1109/TEVC.2004.826071

[19] Liu J Y, Guo M Z, Deng C, "Particle swarm optimization algorithm based on goose", Computer science, vol.33, no. 11, pp. 166-168, 2006. doi:10.3969/j.issn.1002-137X.2006.11.047

[20] Chen H, Zhu Y, "Optimization based on symbiotic multi-species coevolution", Applied Mathematics & Computation, vol. 205, no. 1, pp. 47-60, 2008. doi: 10.1016/j.amc.2008.05.148

[21] Qin Q D, Li R J, "Two-population particle swarm optimization algorithm based on bioparasitic behavior", Control and Decision, vol. 26, no. 4, pp. 548-552, 2011. doi:10.13195/j.cd.2011.04.71.qinqd.025

[22] Yang P, Sun Y M, Liu X L, "The particle swarm optimization algorithm based on bacterial foraging chemotactic operator", Application Research of Computers, vol. 28, no. 10, pp. 3640-3642, 2011. doi:10.3969/j.issn.1001-3695.2011.10.008

[23] Viswanathan G M, Afanasyev V, Buldyrev S V, "Lévy flight search patterns of wandering albatrosses", Nature, vol. 381, no. 6581, pp. 413-415, 1996. doi:10.1038/381413a0

[24] Shlesinger M F, Klafter J, "Lévy Walks Versus Lévy Flights On Growth and Form", Springer Netherlands, vol. 100, pp. 279-283, 1986. doi:10.1007/978-94-009-5165-5_29

[25] Richer T J, Blackwell T M, "The Lévy Particle Swarm", Evolutionary Computation, 2006. CEC 2006. IEEE Congress on. IEEE, 2006, pp. 808-815, 2006. doi:10.1109/CEC.2006.1688394

[26] Wang D, Tang C Q, Tian B G, Qu L S, Zhang J C, Di Z R, "The Lévy flight and Brownian motion characteristic cycle of competition game and stable species coexistence conditions", Acta Physica Sinica, vol. 16, pp. 439-446, 2014. doi:10.7498/aps.63.168701

[27] Li D, "Cooperative quantum-behaved particle swarm optimization with dynamic varying search areas and Lévy flight disturbance", The scientific world journal, vol. 2014, 2014. doi:10.1155/2014/370691

[28] Yan X F, Ye D Y, "An improved algorithm of bacteria foraging based on the Lévy flight", Computer Systems & Applications, vol. 24, no. 3, pp. 124-132, 2015. doi:10.3969/j.issn.1003-3254.2015.03.021

[29] Haklı H, Uğuz H, "A novel particle swarm optimization algorithm with Lévy flight", Applied Soft Computing, vol. 23, no. 5, pp. 333,345, 2014. doi:10.1016/j.asoc.2014.06.034

[30] Martín D, Caballero B, Haber R, "Optimal Tuning of a Networked Linear Controller Using a Multi-Objective Genetic Algorithm. Application to a Complex Electromechanical Process", International Journal of Innovative Computing Information & Control Ijicic, vol.5, pp. 3405-3414, 2009. doi: 10.1109/ICICIC.2008.407

[31] Harmanani H M, Drouby F, Ghosn S B, "A parallel genetic algorithm for the open-shop scheduling problem using deterministic and random moves", International Journal of Artificial Intelligence, vol. 14, no. 1, pp. 130-144, 2016. doi: 10.1145/1639809.1639841

[32] Castillo O, Neyoy H, Soria J, "A new approach for dynamic fuzzy logic parameter tuning in Ant Colony Optimization and its application in fuzzy control of a mobile robot", Applied Soft Computing Journal, vol. 28, pp. 150-159, 2015. doi: /10.1016/j.asoc.2014.12.002

[33] Precup R E, Sabau M C, Petriu E M, "Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for anti-lock braking systems", Applied Soft Computing, vol. 27, pp. 575-589, 2015. doi: 10.1016/j.asoc.2014.07.004

[34] Jensi R, Jiji G W, "An enhanced particle swarm optimization with Lévy flight for global optimization", Applied Soft Computing, vol. 43, pp. 248-261, 2016. doi:10.1016/j.asoc.2016.02.018

[35] Manesh M H K, Ameryan M, "Optimal design of a solar-hybrid cogeneration cycle using Cuckoo Search algorithm", Applied Thermal Engineering, vol. 102, pp. 1300-1313, 2016. doi:10.1016/j.applthermaleng.2016.03.156

[36] Suganthan P N, Hansen N, Liang J J, et al. "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization". Nanyang Technological University, 2005, 1-50.

[37] Mallipeddi R, Suganthan P N. "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization". Nanyang Technological University, 2010, 1-17.