Received December 1, 2019, accepted January 9, 2020, date of publication January 27, 2020, date of current version March 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969526

Swarm Optimization Improved BP Algorithm for Microchannel Resistance Factor

TENG SHEN, JIAQING CHANG, AND ZHONGWEI LIANG
School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China

Corresponding author: Jiaqing Chang (cjq2018@163.com)

This work was supported in part by the Guangzhou Science and Technology Plan Project under Grant 201904010491, and in part by the Science and Technology Novel Research Program in Higher Educational Universities of Guangdong Province under Grant 2017KCXTD025.

ABSTRACT In this paper, a new swarm optimization improved BP (Back Propagation) algorithm, combination of PSE (Particle Swarm Evolution) and BP, called PSE-BP algorithm, is introduced to train ANN (Artificial Neural Network) for the purpose of microchananel resistance factor prediction. The PSE algorithm was firstly proposed by comprehensively learning the principle of gradient descent, genetic algorithm and particle swarm optimization. Then, the search capability of PSE was analyzed by a standard cost function. Its appropriate control parameters were also determined by the same time. By utilizing the global search ability and high search efficiency of PSE algorithm, the improved BP (PSE-BP) algorithm was proposed. The training efficiency of PSE-BP algorithm was tested and compared with BP algorithm using Iris dataset. Finally, the resistance factor of rectangular cross-section microchannel was established using ANN, and trained with PSE-BP and BP algorithm, respectively. The results show that the PSE-BP algorithm can greatly improve the training efficiency of ANN, compare with BP algorithm. And the microchananel resistance factors, which predicted by ANN model and trained with PSE-BP algorithm, are in good agreement with the simulation samples.

INDEX TERMS Particle swarm evolution, BP, ANN, training efficiency, resistance, microchannel.

I. INTRODUCTION

Artificial neural network (ANN) is a kind of mathematical model, which is the simulation and simplification of the information processing and transmission mechanism of biological nervous system. ANN has extremely strong adaptability to nonlinear problems by training to approach target system, which is also called learning [1]–[4]. The training process of ANN is actually the minimum optimization process of the output error function (cost function), and using all the weights and biases as argument. It is important to provide accurate numerical methods to predict flow characteristics such as pressure drop and resistance factor of fluid in microchannel. Based on the strongly nonlinear recognition ability, ANN has been widely used in microfluidic devices such as Bar *et al.* [5] studied the non-Newtonian liquid flow through piping components using ANN. Alizadehdakhel *et al.* [6] studied the two-phase flow pressure drop using ANN. Zhao and Su [7] predicted the pressure coefficient of cyclone separators using ANN. Rahimi *et al.* [8] predicted the flow characteristic in serpentine micro-channels by the application of ANN and GA (genetic algorithm). Beigzadeh and Rahimi [9] studied the heat transfer and flow characteristics in helically coiled tubes using ANN. Moreover, Xiea *et al.* [10], Rosa *et al.* [11] Cai *et al.* [12], Wu *et al.* [13] studied the flow regime classification using ANN. All the researches showed that the ANN has very high accuracy in the application of prediction and classification.

Theoretically, all the function optimization algorithms can be adopted to train ANN. A back propagation (BP) algorithm [14]–[16] and krill herd algorithm(KHA) [17], [18] are the most widely used ANN training algorithm at present. The BPN(BP algorithm trained neural network) has been widely applied in many fields, such as pattern recognition [19], function approximation [20] and image processing [21] etc. However, it was found that there were many shortcomings in the training process of BPN, including slow convergence, the emergence of local extremum, low learning efficiency

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

and so on. In this case, many improvements have been done to overcome the shortcomings of BP algorithm, such as additional momentum method [22], adaptive learning rate method, and flexible BP algorithm [23]. Due to BP algorithm is based on gradient descent (GD) method. It is actually using one particle to search the solution space of the output error function. Therefore, that improvement method still can't overcome the local optimization problem of BP algorithm [24]. There is no doubt that it is easy to converge to local optimum when the output error function has multiple minimum values.

To avoid local optimum problem of BP algorithm, swarm optimization algorithm was adopted to train ANN. The most widely used swarm optimization algorithms are genetic algorithm (GA) and particle swarm optimization (PSO) algorithm. Bui and Hasegawa [25] applied the hybrid model including BPN and genetic algorithm (GA) to estimate the nanofluids density. Valdez et al. [26] designed a new approach using fuzzy logic to dynamically adapt some parameters of the particle swarm optimization algorithm, such as the weight of inertia and learning factors. Yu and Xu [27] studied the weld appearance prediction with BP neural network improved by genetic algorithm. Moreover, Zhang *et al.* [28] proposed a method to optimize a BP algorithm based on genetic algorithm (GA) to speed the training of BP, and to overcome BP's disadvantage of being easily stuck in a local minimum.GA utilizes a large number of individuals to search the solution space of the cost function to simulate the biological gene evolution process. There is no doubt that GA has good global searching ability and can learn the near-optimum solution without the gradient information of error functions. But it has weak global search ability and slow convergence speed.

Furthermore, many researchers also [29]–[31] tried to use PSO to train ANN by adding an extra term to the velocity update function reduced the possibility of sticking in a local minimum. As is known to all, PSO is simulating the birds foraging process by tracking the history optimum of the particle itself and the particle swarm to search the solution space of the cost function, it has the characteristic of fast convergence speed and can solve some complex problems [32]. However, due to the historical optimum tracking mechanism, it is still possible to converge to local optimum.

The goal of this paper is to study swarm optimization algorithm to improve the BP algorithm, overcome the local optimum problem of BP algorithm and improve the training efficiency of ANN for microchannel resistance factor. By analyzing, we know that GD algorithm has strong local search ability but weak global search ability, GA algorithm has strong global search ability but weak local search ability, and PSO algorithm has fast convergence speed but weak reliability. Therefore, GD, GA and PSO have the characteristic of complementary advantages. By the combination of GA, GA and PSO, a novel swarm optimization algorithm was proposed, which called particle swarm evolution (PSE) algorithm. Then, the BP algorithm is improved to propose PSE-BP algorithm, and the efficiency of BP-PSE and BP algorithm are also contrasted by Iris dataset and microchannel resistance sample.

II. PSE ALGORITHM

A. ANALYSIS OF GD, GA AND PSO ALGORITHMS

For the PSE algorithm is based on the GD, GA and PSO algorithms, so we will introduce their principles first.

(1) The basic principle of GD algorithm is searching the minimum value along the negative gradient direction of the cost function. By continuously updating the position of the search particle in the solution space of the cost function, GD can reach the minimum finally. Due to the negative gradient direction is the fastest decrease direction of the cost function value, so GD is also called steepest descent algorithm.

Assuming that the target function is $y = f(x)$, how to search the parameter value x when y is equal to the expected value $\bar{y}$. The cost function of this question can be expressed as

$$e(x) = \frac{1}{2}(y - \bar{y})^2 \tag{1}$$

Using GD, the position (x) update operator can be expressed as

$$x(k+1) = x(k) - \eta \left.\frac{de(x)}{dx}\right|_{x=x(k)} \tag{2}$$

where, k is the searching time, η is learning rate.

(2) GA algorithm utilizes three kinds of genetic operations, which are crossing, variation and selection, to drive the population evolve to higher sufficiency. For real encoding, the crossing operator of GA can be expressed as

$$x_c = r x_{p1} + (1 - r) x_{p2} \tag{3}$$

where, r is a random value between 0 to 1, x_{p1} and x_{p2} are the gene (position of the search particle) of the parents respectively, and x_c is gene of the child.

In GA algorithm, the parents is selected by using roulette algorithm based on the fitness of the individuals in the population. The higher the individual's fitness, the higher the selected proportion of the parents. And the fitness function to compute the fitness of the gene of the individual can be expressed as

$$g(e_i) = \begin{cases} \dfrac{\hat{e} - e_i}{\sum\limits_{j=1}^{N} (\hat{e} - e_j)} & \max\{e_i\} \neq \min\{e_i\} \\ \dfrac{1}{N} & \max\{e_i\} = \min\{e_i\} \end{cases} \tag{4}$$

where, $\hat{e}$ can be expressed as

$$\hat{e} = \max\{e_i\} + 0.001 * [\max\{e_i\} - \min\{e_i\}] \tag{5}$$

After crossing, the individuals with the minimum fitness will be eliminated, which is called selection. And the minimum fitness of some residual individuals will take different operation. The variation operator of GA can be expressed as

$$x_{new} = C_v r x_{old} \tag{6}$$

where, C_v is a constant, normally set to 2.

(3) PSO algorithm has only two kinds of operations, which are speed update and position update. The speed update operator can be expressed as

$$v(k+1) = v(k) + c_1 r_1 \left[p(k) - x(k)\right] + c_2 r_2 \left[g(k) - x(k)\right] \tag{7}$$

where, c_1 and c_2 are constants, r_1 and r_2 are random values between 0 to 1, $p(k)$ is the history optimum of the particle itself, $g(k)$ is the history optimum of the particle swarm. The position operator of PSO can be expressed as

$$x(k+1) = x(k) + v(k+1) \tag{8}$$

In this way, the optimum can be searched by continuously adjusting the speed and position of the particles in the particle swarm.

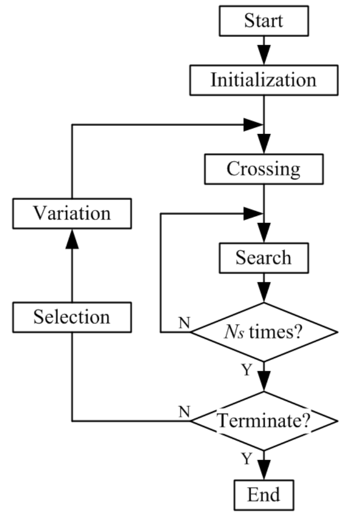


FIGURE 1. **Searching process of PSE.**

B. PRINCIPLE OF PSE ALGORITHM

PSE algorithm is not a simple combination of PSO, GA and GD algorithm, but an improvement of GD algorithm based on the core features of PSO's group search and GA's population evolution. GD algorithm uses a single particle to search the solution space of the cost function. When the solution space has multiple minima, it is hard to converge to the minimum. At this time, by adopting the group search mechanism of PSO, using multiple GD search particles to parallel search the solution space, which will greatly improve the probability of finding the global optimal point. However, due to the inherent characteristics of GD algorithm (searching in the direction of negative gradient), the algorithm still has a high risk of falling into local minimum. In an extreme case, when all search particles fall into the same local optimal region, the GD algorithm can only converge to the local optimal point. It has no ability to jump out of the local optimal region because of the negative gradient search characteristics. At this time, combining the mutation mechanism of GA algorithm (multiplying the position value with a larger random number), giving the search particles jumping ability to depart from the local optimal region. Meanwhile, the GA's crossover mechanism derives new search particles by using comparative advantage search particles. The new search particles inherit the advantages of the original search particles can improve the convergence speed of the algorithm. And the GA's selection mechanism, by eliminating inefficient search particles, can make more efficient use of computer computing resources.

The principle of PSE algorithm is based on PSO algorithm, which also utilizes a large number of particles to form particle swarm and then searches the solution space of the cost function. Different from PSO algorithm, the speed update of the PSE algorithm does not adopt the way of tracking the historical optimal position, but the gradient descent method which increases the inertia term. As same as GD, the search speed of the particle is increased along the negative gradient direction of the cost function. And, the search acceleration of particle is generated by the gradient of cost function. Therefore, the speed update operator of PSE can be express as

$$v(k+1) = \alpha v(k) - \beta \left.\frac{\partial e(x)}{\partial x}\right|_{x=x(k)} \tag{9}$$

where, α is velocity attenuation factor; β is the acceleration regulator factor.

The position update operator of PSE is the same as PSO, refer to Eq(8). After several times of searching, the particle swarm will be carried out a genetic operation to eliminate the particles with small fitness value, retain the strong fitness particles and generate new particles. The genetic operators of PSE are the same as GA, which shown in Eq(3) to Eq(6).

The searching process of PSE is shown in Fig.1, which N, P_c, P_v and N_s are the population of the particle swarm, the crossover probability, the variation probability and the searching times, respectively. The specific operation process of PSE is as follows.

(1) Initialization. N particles were generated first, then the position and velocity of these particles were set to random value and zeros.

(2) Crossing. Calculating the fitness value of particles in the particle swarm, refer to Eq(4). According to the fitness value of the particles, using the roulette wheel algorithm to select two particles each time to crossover and then generate new particles, refer to Eq(3). Repeat $P_c N$ times.

(3) Search. Updating the velocity and position of the particles in the particle swarm, refer to Eq(9) and Eq(8). Repeat N_s times.

(4) Termination judgment. Calculating the fitness value of the particles in the particle swarm, refer to Eq(4). If the value of cost function of the biggest fitness particles is less than the convergence value, end the process. If not, go on to the next step.

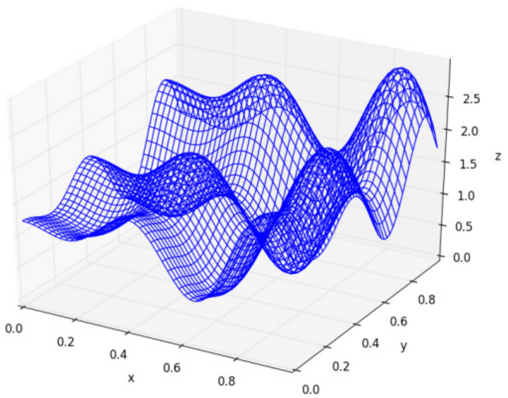


FIGURE 2. Picture of the cost function.

(5) Selection. Calculating fitness value of each in particle swarm, and eliminating P_cN particles with the smallest fitness value.

(6) Variation. Selecting P_cN particles with minimum fitness in particle swarm for variation operation, refer to Eq(6). And go to step (2).

III. SEARCH CAPABILITY ANALYSIS OF PSE

The control parameters of the PSE algorithm include N, P_c, P_v, N_s, α and β. Next, we will employ a standard cost function (Eq3.51) to study the capability of PSE by the control parameters. And the picture of cost function is shown in Fig 2.

$$z = 1.24 - x\sin(4\pi x) + y\sin(4\pi y + \pi) \tag{10}$$

where, $x \in [0, \quad 1]$; $y \in [0, \quad 1]$. In our calculation, the max evolution times $N_{e\max}$ and convergence error $e_{\max}$ were set to 100 and 0.0001. By comparing the proportion of convergence times (search success rate R_s) in 100 tests under different control parameters, the average evolution generation N_{eq} and the maximum evolution generations N_{max} converge to test the searching ability of the PSE algorithm. It means that the higher the search success rate is, the smaller the average and the maximum evolution generations are, and the stronger the search ability of the algorithm is. The values of the control parameters and test results are shown in Tab.1.

The curves of search capability varying with control parameters are shown in Fig3 - Fig8. Fig 3 shows the test result curve corresponding to different particle swarm optimization scale from Num1, 2, 3, 4 and 5 in Table 1. It is observed that with increase of particle swarm size, the search success rate of algorithm remains unchanged at 100%, and the average and the maximum evolution generations decrease gradually, but the amplitude of the decrease is reduced. The results show that increasing the particle swarm size can improve its search ability, but when the particle swarm size exceeds a certain limit, continuing to increase the population size does not contribute much to improving the search capability of the PSE algorithm.

TABLE 1. The values of the control parameters and test results.

N_o	N	P_c	P_v	α	β	N_s	N_{eq}	$N_{\max}$	R_s
1	2	0.5	0.5	0.1	0.00001	20	36.95	81	100%
2	4	0.5	0.5	0.1	0.00001	20	23.99	68	100%
3	8	0.5	0.5	0.1	0.00001	20	14.92	47	100%
4	16	0.5	0.5	0.1	0.00001	20	10.08	33	100%
5	32	0.5	0.5	0.1	0.00001	20	6.84	19	100%
6	4	0	0.5	0.1	0.00001	20	57.07	93	98%
7	4	0.25	0.5	0.1	0.00001	20	27.43	74	100%
8	4	0.75	0.5	0.1	0.00001	20	21.94	98	100%
9	4	1	0.5	0.1	0.00001	20	16.73	71	100%
10	4	0.5	0	0.1	0.00001	20	52.14	99	43%
11	4	0.5	0.25	0.1	0.00001	20	24.98	72	98%
12	4	0.5	0.75	0.1	0.00001	20	24.71	74	100%
13	4	0.5	1	0.1	0.00001	20	50.5	98	14%
14	4	0.5	0.5	0	0.00001	20	25.48	69	100%
15	4	0.5	0.5	0.5	0.00001	20	18.17	56	100%
16	4	0.5	0.5	0.9	0.00001	20	10.28	41	100%
17	4	0.5	0.5	1	0.00001	20	5.72	27	100%
18	4	0.5	0.5	0.1	0.0001	20	8.46	35	100%
19	4	0.5	0.5	0.1	0.001	20	2.15	8	100%
20	4	0.5	0.5	0.1	0.01	20	1.37	6	100%
21	4	0.5	0.5	0.1	0.1	20	53.53	95	19%
22	4	0.5	0.5	0.1	0.00001	5	25.12	98	95%
23	4	0.5	0.5	0.1	0.00001	10	28.62	89	98%
24	4	0.5	0.5	0.1	0.00001	30	18.85	61	100%
25	4	0.5	0.5	0.1	0.00001	40	18.77	44	100%

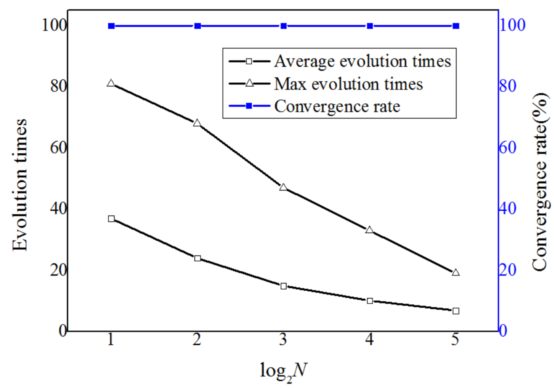


FIGURE 3. Curve of search capability varying with particle swarm optimization scale.

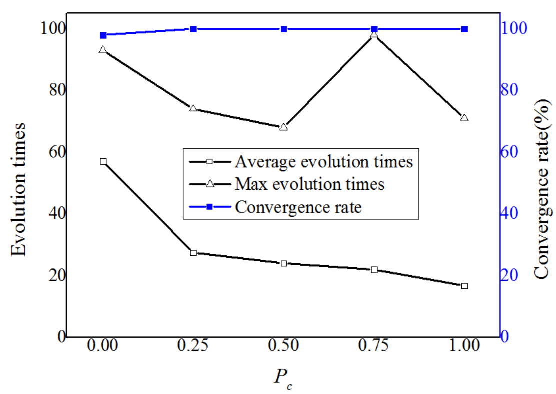


FIGURE 4. Curve of search capability varying with crossover probability.

Fig 4 shows the test result curve corresponding to different crossover probability from Num6, 7, 2, 8 and 9 in Table 1. As you can see, the search success rate of the algorithm is 100% except when the crossover probability is zero. With the increase of crossover probability, the average evolution

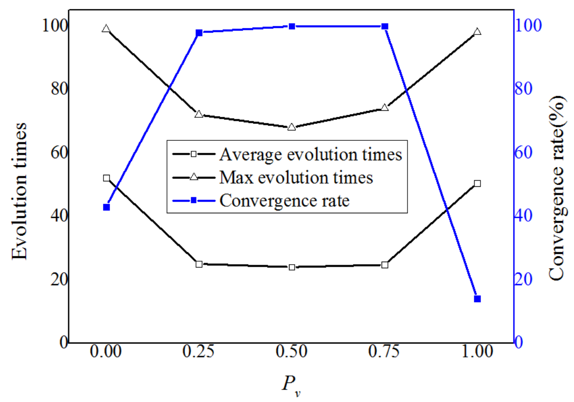


FIGURE 5. Curve of search capability varying with variation probability.

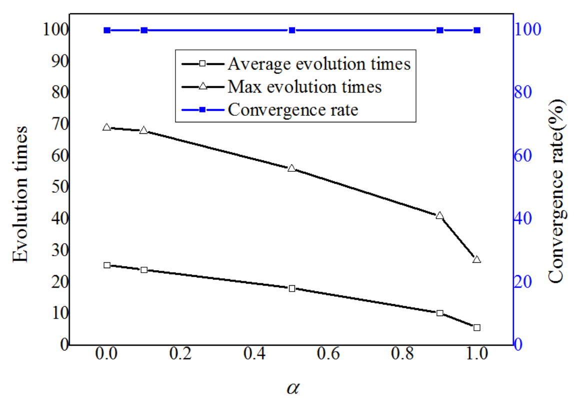


FIGURE 6. Curve of search capability varying with rate attenuation factor.

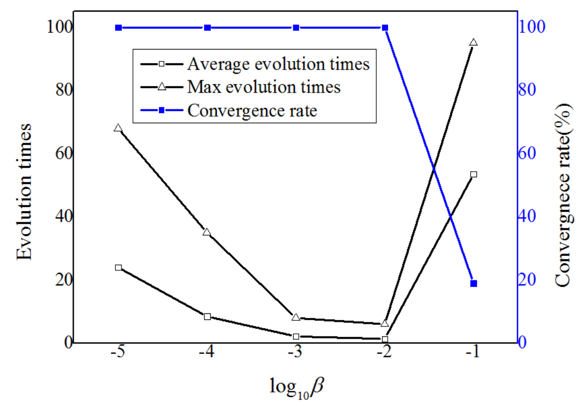


FIGURE 7. Curve of search capability varying with acceleration regulation factor.

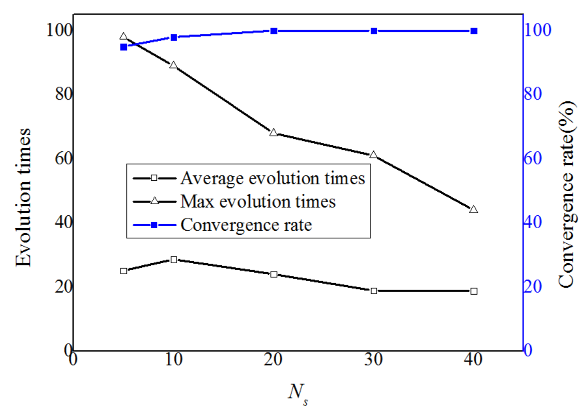


FIGURE 8. Curve of search capability varying with particle search times.

generations decreases gradually, but the maximum evolution generations curve has an oscillation variety. In general, crossover operation can improve the search ability of the algorithm. However, the excessive crossover probability can cause the algorithm to oscillate easily which results in non-convergence. As a result, it is appropriate to set the crossover probability at 0.5.

Fig 5 shows the test result curve corresponding to different variation probability from Num 10, 11, 2, 12 and 13 in Table 1. Obviously, with the increase of mutation probability, the average and maximum evolution generations decrease first and then increase gradually. However, with the increase of mutation probability, the search success increases first and then decreases. The results show that mutation probability will reduce the search ability of the PSE algorithm no matter larger or smaller, so it should be set at 0.5.

Fig 6 shows the test result curve corresponding to different rate attenuation factor from Num 14, 2, 15, 16 and 17 in Table 1. As you can see, with the decrease of rate attenuation factor, the search success rate of the algorithm is always 100%, but the average and maximum evolution generations tend to decrease gradually. In general, the results show the search ability of the algorithm can be improved by increasing the rate attenuation factor of the PSE algorithm.

Comparison of the test results of group 2, 18, 19, 20 and 21, we get the curve corresponding to different acceleration regulation factor, as shown in Fig7. Apparently, the search success rate remains at 100% when the acceleration regulation factor is less than negative 2, and then it will drop dramatically with increasing the factor. Actually, when the acceleration regulation factor is 0.1, the search success is really less than 20%. Furthermore, with increasing the factor, the average and maximum evolution generations tend to decrease and then increase abruptly. In general, an appropriate acceleration regulation factor can improve the search ability of the algorithm, but larger factor will cause the algorithm to oscillate which decreases the searching capability. Therefore, in order to ensure the searching efficiency, the acceleration regulation factor should be set restrainedly.

Fig8 shows the test result curve corresponding to different particle search times from Num22, 23, 2, 24 and 25 in Table 1. It is observed that with the increase of particle search times, the search success rate of algorithm remains unchanged at 100%, and the average and the maximum evolution generations decrease gradually, but the amplitude of the decrease is reduced. The results show that increasing the particle search times can improve the search speed of the PSE algorithm, but too high particle search times has little contribution to improving the search ability of the algorithm.

In summary, the search ability of the PSE algorithm is affected by the population size, crossover probability, mutation probability, velocity attenuation factor, acceleration regulation factor, and particle search times. The larger the population size, acceleration regulation factor, and particle search times, the stronger the search ability of the algorithm. Then the crossover probability and mutation probability are set to 0.5, too large or too small will reduce the search ability of the algorithm. In addition, it should be noted that the acceleration regulation factors must be set carefully. If the acceleration regulation factor is too small, the search speed of the algorithm is slow, but if the acceleration is too large, the algorithm may oscillate without convergence.

IV. PSE-BP ALGORITHM

From the above analysis, we can see that the PSE algorithm has the characteristics of strong ability of searching for global and local and it also has fast searching speed that can greatly improve the efficiency of neural network training. It is necessary to solve the gradient of the error function for each layer weight matrix and offset vector because the particle velocity of the PSE algorithm is related to the gradient of the error function, but the BP algorithm has obvious advantages in solving the corresponding gradient. According to this feature, combining the PSE algorithm with the BP algorithm and then the PSE-BP algorithm is obtained. In this way, the corresponding gradient can be solved by the BP algorithm, and then the PSE algorithm is used to update each layer weight matrix and offset vector.

The basic variables of the neural network are the weight matrix and offset vector. The position of the particles is represented in their set form. So, for n+1 feed-forward neural network, the position of the particles can be expressed as:

$$x = \{\mathbf{W}_1, \ldots, \mathbf{W}_n, B_1, \ldots, B_n\} \tag{11}$$

The speed of the particles can be expressed as:

$$v = \{\mathbf{V}_{\mathbf{W}1}, \ldots, \mathbf{V}_{\mathbf{W}n}, V_{B1}, \ldots, V_{Bn}\} \tag{12}$$

The particle velocity update operator can be expressed as:

$$\mathbf{V}_{\mathbf{W}i}(k+1) = \alpha \mathbf{V}_{\mathbf{W}i}(k) - \beta \left.\frac{\partial e}{\partial \mathbf{V}_{\mathbf{W}i}}\right|_{\mathbf{V}_{\mathbf{W}_i} = \mathbf{V}_{\mathbf{W}i}(k)} \tag{13}$$

$$V_{Bi}(k+1) = \alpha V_{Bi}(k) - \beta \left.\frac{\partial e}{\partial B_i}\right|_{B_i = B_i(k)} \tag{14}$$

The particle position update operator can be expressed as:

$$\mathbf{W}_i(k+1) = \mathbf{W}_i(k) + \mathbf{V}_{\mathbf{W}i}(k+1) \tag{15}$$

$$B_i(k+1) = B_i(k) + V_{Bi}(k+1) \tag{16}$$

The crossover operator can be expressed as:

$$\mathbf{W}_i^c = \mathbf{R}\mathbf{W}_i^{p1} + (\mathbf{D} - \mathbf{R})\mathbf{W}_i^{p2} \tag{17}$$

$$B_i^c = RB_i^{p1} + (D - R)B_i^{p2} \tag{18}$$

TABLE 2. Lable definition of setosa versicolor virginica.

	C_{set}	C_{ver}	C_{vir}
Setosa	1	0	0
Versicolor	0	1	0
Virginica	0	0	1

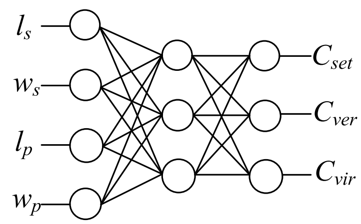


FIGURE 9. Topology of the Iris classification ANN.

where, **R** and **D** are a random matrix between 0 to 1 and all 1 matrix, which has the same values of dimensions of $\mathbf{W}_i$. And R and D are a random vector between 0 to 1 and all 1 vector, which has the same values of dimensions of B_i. The mutation operator can be expressed as:

$$\mathbf{W}_i^{new} = \mathbf{R}\mathbf{W}_i^{old} \tag{19}$$

$$B_i^{new} = RB_i^{old} \tag{20}$$

V. TRAINING EFFICIENCY TEST OF PSE-BP

In order to verify the efficiency of the PSE-BP algorithm for neural network training, we used the Iris standard data set to make comparison tests on training efficiency of the PSE-BP algorithm and the BP algorithm. The Iris standard data set is a widely used pattern classification system test data set which includes a total of 150 samples and is divided into Setosa, Versicolor and Virginica. Each type of sample accounts for 1/3 of the total number of samples, which are respectively expressed as C_{set}, C_{ver} and C_{vir}. The lable definition of Setosa Versicolor and Virginica is listed in TABLE 2.

For the Iris has four features and three modes. Therefore, a three-layer feed-forward neural network with 4-3-3 structure is adopted. The network's hidden layer and output layer activation functions are all adopted as Sigmoid functions. The network structure is shown as following Fig 9.

In order to make the network easier to converge, the sample needs to be preprocessed. The sample pretreatment formula can be expressed as:

$$s_i = \frac{c_i - \min(c_i)}{\max(c_i) - \min(c_i)} \tag{21}$$

where s_i, x_i are respectively sample values before and after pretreatment.

The network was trained by using the PSE-BP algorithm, the standard BP algorithm, and the BP algorithm that adds the inertia item. The train settings of the algorithms are listed in TABLE 3. The relationship curve between average output variance of the network by training, evolutionary algebra of the PSE-BP algorithm and the searching times of the BP algorithm are shown as following Fig 10.

TABLE 3. Train settings of PSE-BP standard BP and inertia BP.

	PSE-BP	*Standard BP*	*Inertial BP*
N	20	/	/
P_c	0.5	/	/
P_v	0.5	/	/
α / momentum	0.9	/	0.9
β /learn rate	0.01	0.01	0.01
N_s	50	/	/

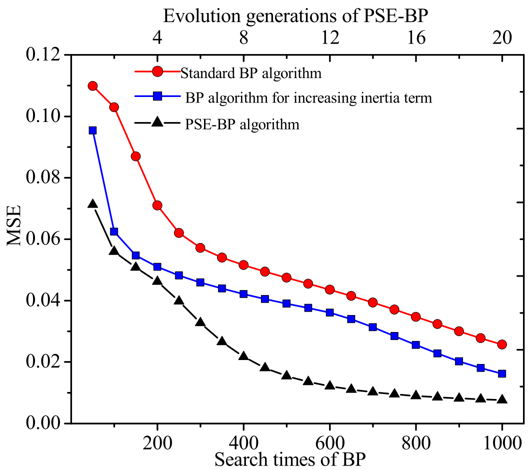


FIGURE 10. Training result of Iris classification ANN.

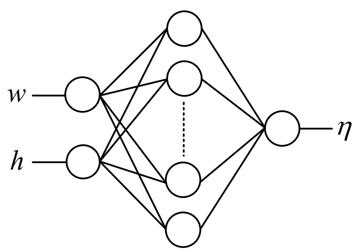


FIGURE 11. Microchannel resistance feature model.

As we can see from the Fig 10, the PSE-BP algorithm is superior to the BP algorithm both in mean squared deviation of training and the decline rate of the mean squared curve. Compared with the standard BP algorithm, the BP algorithm that adds the inertial term has greatly improved its training efficiency, but it is still less than the training efficiency of the PSE-BP algorithm. The results show that the PSE-BP algorithm can improve the convergence speed of multi-layer feed-forward neural network and improve training efficiency.

VI. MICROCHANNEL RESISTANCE FACTOR MODEL

For the rectangle cross-section microchannel, the pressure curve changes liner, so the resistance factor is constant [34]. Therefore, the structure parameters of rectangular cross-section straight micro-channel are the width and height of the micro-channel, so the input and signal are w, h, and η, respectively. Based on ANN, the micro-channel resistance feature model is established, as shown in Fig 11.

TABLE 4. The partial microchannel resistance factor samples.

w (um)	h (um)	η	w (um)	h (um)	η
50	100	2030940000	300	100	477084000
50	150	1775540000	300	150	259410000
50	200	1669180000	300	200	185899000
50	250	1611280000	300	250	151767000
50	300	1574910000	300	300	132049000
50	350	1549950000	300	350	115058000
50	400	1531750000	300	400	102893000
50	450	1517880000	300	450	93767900
50	50	3167510000	300	50	1574910000
50	500	1506970000	300	500	86682200
100	100	891227000	350	100	458557000
100	150	647579000	350	150	240865000
100	200	553454000	350	200	167374000
100	250	505497000	350	250	133990000
100	300	477084000	350	300	115058000
100	350	458557000	350	350	102779000
100	400	445624000	350	400	91101400

Using self-designed automatic simulation framework based on the Fluent CFD, the microchannel width and height ranging from 50 μm to 500 μm, 200 sets of resistance factor samples were obtained, and the partial microchannel resistance factor samples are shown in Tab VI. The detailed simulation model, parameter setting and simulation method have been shown in our previous paper [35].

In order to further compare the training efficiency of BP algorithm and PSE-BP algorithm, the BP algorithm and PSE-BP algorithm were used to train the microchannel resistance factor model. To make the model more convergent, the resistance factor samples were pretreated before training, and the pretreatment formula was shown in Eq.21.

A. BP ALGORITHM TRAINING

The inertial regulatory factor for BP algorithm is 0.9, the learning rate is 0.01, and the convergence error is 0.001. Starting from 1, the number of hidden layer neurons of the model is gradually enlarged, and the model converges when the number of hidden layer neurons is 2. The each layer weight matrix and offset vector obtained by training is shown in Eq.22. And the relation curve between the variance of the network output and the particle search frequency during the training process is shown in Fig.12.

$$\mathbf{W}_1 = \begin{bmatrix} 0.07 & 9.93 \\ 9.12 & 0.02 \end{bmatrix}, B_1 = \begin{bmatrix} 1.01 \\ 0.88 \end{bmatrix},$$
$$\mathbf{W}_2 = \begin{bmatrix} -2.32 \\ -2.17 \end{bmatrix}, B_2 = [4.48] \tag{22}$$

B. PSE-BP ALGORITHM TRAINING

For PSE-BP algorithm, the particle swarm size is 20, the crossover probability is 0.5, the mutation probability is 0.5, the velocity attenuation factor is 0.9, the acceleration adjustment factor is 0.01, the particle search frequency is 20, and the convergence error is 0.001. And also start from 1, the number of hidden layer neurons of the model is gradually

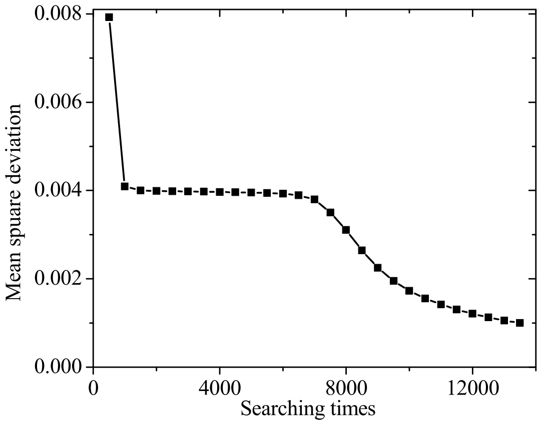


FIGURE 12. Relationship between mean square deviation and evolution generation in BP algorithm training.

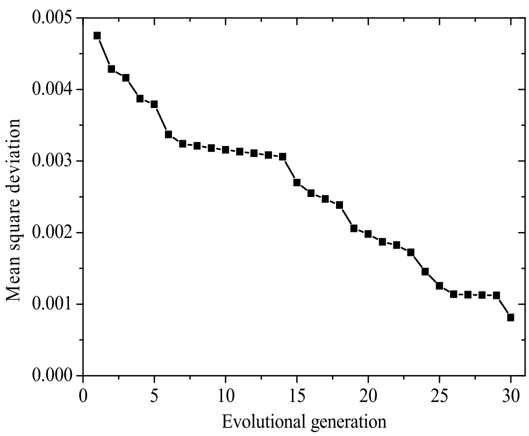


FIGURE 13. Relationship between mean square deviation and evolution generation in PSE-BP algorithm training.

increased, and the model converges when the number of hidden layer neurons is 2. The weight matrix and offset vector obtained by training is shown in Eq.23. The relation curve between the variance of the network output and evolutionary algebra during the training process is shown in Fig 13.

$$\mathbf{W}_1 = \begin{bmatrix} -0.21 & -10.3 \\ -11.83 & 0.19 \end{bmatrix}, B_1 = \begin{bmatrix} -0.97 \\ -1.42 \end{bmatrix},$$
$$\mathbf{W}_2 = \begin{bmatrix} 3.23 \\ 3.04 \end{bmatrix}, B_2 = [0.02] \tag{23}$$

As shown in Fig 12, the model can converge only under the condition of the search frequency of BP algorithm must reach more than 12,000. And the relation curve has a long flat between its variance of the network output and the search times. The training efficiency is low. From Fig 13, the evolutionary algebra of PSE-BP algorithm training is 30 generations, and the search times of equivalent individual particles is only 600. Apparently, the training efficiency is more than 20 times higher than that of the BP algorithm with over 12,000. It is further proved that the PSE-BP algorithm can greatly improve the training efficiency of the microchannel resistance factor model compared with BP algorithm.

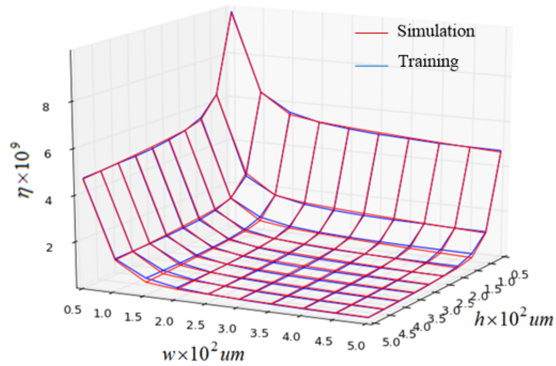


FIGURE 14. Comparison of characteristic factor of microchannel resistance between simulation and training.

The comparison between the microchannel resistance feature coefficients which calculated by the training results of PSE-BP algorithm and the simulation results is shown in Fig.14. It can be seen that the simulation is in good agreement with the training results, and the error is small(within 5%). As a whole, the microchannel resistance factors can be effectively extracted by using the PSE-BP algorithm, and the weight matrix and offset vector obtained by training are correct.

VII. CONCLUSION

In this paper, the principle of gradient descent method, genetic algorithm and particle swarm optimization were studied. With the combination of gradient descent method, genetic algorithm and particle swarm, a novel swarm optimization algorithmcalled particle swarm evolution was proposed. The searchingcapability of PSE was analyzed by using a standard cost function. Results show that, PSE can obtain strong searching capability by setting appropriate parameters, the crossover rate and variationrate of PSE is better to set to 0.5. It also revealedthat the greater population size, searching times and speed factor, the stronger searching capability of PSE is. And, the acceleration regulation factors must be set carefully. If the acceleration regulation factor is too small, the search speed of the algorithm is slow, but if the acceleration is too large, the algorithm may oscillate without convergence. Moreover, increasing the particle search times can improve the search speed of the PSE algorithm, but too high particle search times have little contribution to improving the search ability of the algorithm.

In general, PSE algorithm has the characteristics of strong ability of searching for global and local. It also has fast searching speed that can greatly improve the efficiency of neural network training. Therefore, PSE was used to improve BP algorithm for ANN training process, which called the PSE-BP algorithm. Iris dataset was adopted to test the training efficiency of PSE-BP, and compare with BP algorithm. The result shows that the PSE-BP algorithm is superior to the BP algorithm both in mean squared deviation of training and the decline rate of the mean squared curve, and it can improve the convergence speed of multi-layer feed-forward neural network and training efficiency. Moreover, for microchannel resistance factor model, the training results of PSE-BP algorithm are in good agreement with the simulation samples of microchannel factor, and the error is within 5%. The results show that the PSE-BP algorithm can greatly improve the training efficiency of ANN, compared with BP algorihtm.

In this research, the rectangular cross-section straight micro-channel are adopted to test the suitable of PSE-BP algorithm, and the input signal are only w and h. Therefore, in order to further verify the application of PSE-BP method to resistance factor model, more complex microchannel will be researched in our future work, such as serpentine or curved microchannel.

REFERENCES

[1] K. Suzuki, I. Horiba, and N. Sugie, "A simple neural network pruning algorithm with application to filter synthesis," *Neural Process. Lett.*, vol. 13, pp. 43–53, Feb. 2001.

[2] B. Walker, C. S. Holling, S. R. Carpenter, and A. Kinzig, "Resilience, adaptability and transformability in social-ecological systems," *Ecology Soc.*, vol. 9, no. 2, p. 5, 2004.

[3] Z. Zhao, J. Shi, X. Lan, X. Wang, and J. Yang, "Adaptive neural network control of a flexible string system with non-symmetric dead-zone and output constraint," *Neurocomputing*, vol. 283, pp. 1–8, Mar. 2018.

[4] Z. Zhao, X. Wang, C. Zhang, Z. Liu, and J. Yang, "Neural network based boundary control of a vibrating string system with input deadzone," *Neurocomputing*, vol. 275, pp. 1021–1027, Jan. 2018.

[5] N. Bar, T. K. Bandyopadhyay, M. N. Biswas, and S. K. Das, "Prediction of pressure drop using artificial neural network for non-newtonian liquid flow through piping components," *J. Petroleum Sci. Eng.*, vol. 71, nos. 3–4, pp. 187–194, Apr. 2010.

[6] A. Alizadehdakhel, M. Rahimi, J. Sanjari, and A. A. Alsairafi, "CFD and artificial neural network modeling of two-phase flow pressure drop," *Int. Commun. Heat Mass Transf.*, vol. 36, pp. 850–856, Oct. 2009.

[7] B. Zhao and Y. Su, "Artificial neural network-based modeling of pressure drop coefficient for cyclone separators," *Chem. Eng. Res. Des.*, vol. 88, nos. 5–6, pp. 606–613, May 2010.

[8] M. Rahimi, M. Hajialyani, R. Beigzadeh, and A. A. Alsairafi, "Application of artificial neural network and genetic algorithm approaches for prediction of flow characteristic in serpentine microchannels," *Chem. Eng. Res. Des.*, vol. 98, pp. 147–156, Jun. 2015.

[9] R. Beigzadeh and M. Rahimi, "Prediction of heat transfer and flow characteristics in helically coiled tubes using artificial neural networks," *Int. Commun. Heat Mass Transf.*, vol. 39, no. 8, pp. 1279–1285, Oct. 2012.

[10] T. Xie, S. M. Ghiaasiaan, and S. Karrila, "Artificial neural network approach for flow regime classification in gas–liquid–fiber flows based on frequency domain analysis of pressure signals," *Chem. Eng. Sci.*, vol. 59, no. 11, pp. 2241–2251, Jun. 2004.

[11] E. S. Rosa, R. M. Salgado, T. Ohishi, and N. Mastelari, "Performance comparison of artificial neural networks and expert systems applied to flow pattern identification in vertical ascendant gas–liquid flows," *Int. J. Multiphase Flow*, vol. 36, no. 9, pp. 738–754, Sep. 2010.

[12] S. Cai, H. Toral, J. Qiu, and J. S. Archer, "Neural network based objective flow regime identification in air-water two phase flow," *Can. J. Chem. Eng.*, vol. 72, no. 3, pp. 440–445, Jun. 1994.

[13] H. Wu, F. Zhou, and Y. Wu, "Intelligent identification system of flow regime of oil–gas–water multiphase flow," *Int. J. Multiphase Flow*, vol. 27, no. 3, pp. 459–475, Mar. 2001.

[14] Y. Mi, M. Ishii, and L. H. Tsoukalas, "Flow regime identification methodology with neural networks and two-phase flow models," *Nucl. Eng. Des.*, vol. 204, nos. 1–3, pp. 87–100, Feb. 2001.

[15] J. Leonard and M. A. Kramer, "Improvement of the backpropagation algorithm for training neural networks," *Comput. Chem. Eng.*, vol. 14, no. 3, pp. 337–341, Mar. 1990.

[16] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, Jul. 2004, pp. 985–990.

[17] N. M. Nawi, M. Z. Rehman, and A. Khan, "A new bat based back-propagation (BAT-BP) algorithm," in *Advances in Systems Science*. Cham, Switzerland: Springer, 2014, pp. 395–404.

[18] P. A. Kowalski and S. Łukasik, "Training neural networks with krill herd algorithm," *Neural Process. Lett.*, vol. 44, no. 1, pp. 5–17, Aug. 2015.

[19] A. L. Bolaji, M. A. Al-Betar, M. A. Awadallah, A. T. Khader, and L. M. Abualigah, "A comprehensive review: Krill herd algorithm (KH) and its applications," *Appl. Soft Comput.*, vol. 49, pp. 437–446, Dec. 2016.

[20] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3367–3375.

[21] P. Wang, W. Du, M. Liang, H. Wang, and W. Wei, "Prediction model of total organic carbon content on hydrocarbon source rocks in coal measures established by BP neural network based on logging parameters," in *Proc. 7th Int. Conf. Environ. Eng. Geophys. Summit Forum Chin. Acad. Eng. Eng. Sci. Technol.*, 2016, pp. 234–237.

[22] L. Liu, Y. Hua, Q. Zhao, H. Huang, and A. C. Bovik, "Blind image quality assessment by relative gradient statistics and adaboosting neural network," *Signal Process., Image Commun.*, vol. 40, pp. 1–15, Jan. 2016.

[23] D. Qun, C. Songcan, and Z. Benzhu, "Improved CBP neural network model with applications in time series prediction," *Neural Process. Lett.*, vol. 18, no. 3, pp. 217–231, Dec. 2003.

[24] S. C. Ng, S. H. Leung, and A. Luk, "Fast convergent generalized back-propagation algorithm with constant learning rate," *Neural Process. Lett.* vol. 9, pp. 13–23, Feb. 1999.

[25] N. T. Bui and H. Hasegawa, "Training artificial neural network using modification of differential evolution algorithm," *Int. J. Mach. Learn. Comput.*, vol. 5, no. 1, pp. 1–6, Feb. 2015.

[26] F. Valdez, J. C. Vazquez, P. Melin, and O. Castillo, "Comparative study of the use of fuzzy logic in improving particle swarm optimization variants for mathematical functions using co-evolution," *Appl. Soft Comput.*, vol. 52, pp. 1070–1083, Mar. 2017.

[27] F. Yu and X. Xu, "A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network," *Appl. Energy*, vol. 134, pp. 102–113, Dec. 2014.

[28] Y. Zhang, X. Gao, and S. Katayama, "Weld appearance prediction with BP neural network improved by genetic algorithm during disk laser welding," *J. Manuf. Syst.*, vol. 34, pp. 53–59, Jan. 2015.

[29] S. Ding, C. Su, and J. Yu, "An optimizing BP neural network algorithm based on genetic algorithm," *Artif. Intell. Rev.*, vol. 36, no. 2, pp. 153–162, Feb. 2011.

[30] G. Haibing, G. Liang, and Z. Chi, "Particle swarm optimization based algorithm for neural network learning," *Acta Electronica Sinica*, vol. 32, no. 9, pp. 1572–1574, 2004.

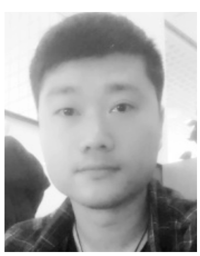
[31] J.-R. Zhang, J. Zhang, T.-M. Lok, and M. R. Lyu, "A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training," *Appl. Math. Comput.*, vol. 185, no. 2, pp. 1026–1037, Feb. 2007.

[32] F. Olivas, L. Amador-Angulo, J. Perez, C. Caraveo, F. Valdez, and O. Castillo, "Comparative study of type-2 fuzzy particle swarm, bee colony and bat algorithms in optimization of fuzzy controllers," *Algorithms*, vol. 10, no. 3, p. 101, Aug. 2017.

[33] M. A. Cavuslu, C. Karakuzu, and F. Karakaya, "Neural identification of dynamic systems on FPGA with improved PSO learning," *Appl. Soft Comput.*, vol. 12, no. 9, pp. 2707–2718, Sep. 2012.

[34] P. Hrnjak and X. Tu, "Single phase pressure drop in microchannels," *Int. J. Heat Fluid Flow*, vol. 28, no. 1, pp. 2–14, Feb. 2007.

[35] T. Shen, J. Chang, J. Xie, and L. Huang, "Analysis of microchannel resistance factor based on automated simulation framework and BP neural network," *Soft Comput.*, vol. 24, no. 5, pp. 3379–3391, May 2019.



TENG SHEN received the B.E. degree from the Liaoning University of Science and Technology, China, in 2012, and the Ph.D. degree from the Nanjing University of Science and Technology, China, in 2017. He is currently pursuing the joint Ph.D. degree with the University of Western Ontario, Canada. He is also a Lecturer with Guangzhou University, China. He has published more than ten research articles in the area of microfluidics and soft computing. His current research interests include application of microfluidic wearable electronic devices and neural networks in inertial microfluidics.

JIAQING CHANG received the B.S. degree in mechanical engineering from Hunan University, Changshan, China, in 2008, the M.S. degree in mechanical engineering from the Beijing University of Technology, Beijing, China, in 2011, and the Ph.D. degree from the Harbin Institute of Technology, China, in 2017. He is currently a Lecturer with Guangzhou University, China. His current research interests include application of micro-droplet, 3D printing, micro-fluid driving, and control technology.

ZHONGWEI LIANG received the B.S. and M.S. degrees in mechatronic engineering from the Shenyang University of Technology, China, in 2000 and 2005, respectively, and the Ph.D. degree in machinery manufacturing and automation from the South China University of Technology, China, in 2008, where he received a postdoctoral certificate, in 2014.

He served as an Academic Research Fellow and a Visiting Research Professor at Zhejiang University, China, The University of New South Wales, Australia, and Northwestern University, USA, from 2014 to 2018. He also served as an Academic Leader of innovative research groups of Guangdong universities and Guangzhou educational system. He is Guangzhou Scholar Distinguished Professor of mechanical engineering, a Supervisor of Ph.D. and Postdoctoral Fellow, and the Deputy Director of the Guangdong Engineering Research Center for strengthen grinding and high performance machining, at Guangzhou University. He has published more than 90 articles in journals and international conferences, simultaneously more than 40 technical patents have been applied and authorized also. His research interests include manufacturing mechanics of AWJ grinding, strengthen grinding, turbulence modeling, fluid mechanics, and advanced manufacturing technology. He is an Associate Editor, Co-Chair, and Reviewer of more than 20 journals and 20 international conferences.