IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 5, 2018, accepted January 4, 2019, date of publication January 23, 2019, date of current version February 27, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2894301*

# A Novel Method for Image Segmentation Based on Simplified Pulse Coupled Neural Network and Gbest Led Gravitational Search Algorithm

**KEMING JIAO AND ZHONGLIANG PAN**
School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou 510000, China
Corresponding author: Zhongliang Pan (panzhongliang@m.scnu.edu.cn)

This work was supported in part by the Guangdong Provincial Natural Science Foundation of China under Grant 2014A030313441, in part by the Guangzhou Science and Technology Project, in part by the Guangdong Province Science and Technology Project under Grant 2016B090918071 and Grant 2014A040401076, and in part by the National Natural Science Foundation of China under Grant 61072028.

**ABSTRACT** This paper proposed a novel image segmentation method based on simplified pulse coupled neural network (SPCNN), which was optimized by gbest led gravitational search algorithm (GLGSA) that combined gravitational search algorithm (GSA) with gbest agent memory ability. To evaluate the performance of GLGSA, we applied it to 23 standard benchmark functions and compared with GSA and GGSA. The results showed that the GLGSA had better performance in term of convergence and avoidance of local minima. Besides, in order to improve the accuracy of segmentation, the fitness function consisted of cross entropy parameter, edge matching, and noise control. To verify the efficiency of our method, we compared it with the state-of-the-art algorithms, such as Otsu, GA Renyi, and PSO-PCNN, using the gray nature images from the Berkeley segmentation dataset. Finally, the subjective visual analysis and quantitative analysis that included the uniformity measure, region contrast measure, structural similarity, and comprehensive evaluation were used to evaluate the segmented images. The comparison results demonstrated that our proposed method could get better segmentation results.

**INDEX TERMS** Image segmentation, gravitational search algorithm, pulse coupled neural network, edge matching, noise control.

## I. INTRODUCTION

Image segmentation is a process that partitions an image into specific, homogeneous regions, which is an imperative and fundamental step in the research field of image analysis, pattern recognition and computer vision. The quality of image segmentation directly affects the image processing later that is considered as the central task in many research fields. So numerous segmentation algorithms have been proposed in recent decades that includes threshold based algorithm [1], region based algorithm [2], edge based algorithm [3], fuzzy theory based [4], neural network based algorithm [5].

Threshold based algorithm is to select a suitable threshold by some criteria which enables it to separate the target from the background. Gritzman *et al.* proposed an adaptive method for lip segmentation and chose the histogram threshold based on the feedback of shape information, which reduced the unnecessary overhead by comparing the reference lip shape model to the initial segmentation [6]. He *et al.* [7] used two-dimensional entropy method for double threshold segmentation, using parallel genetic simulated annealing algorithm to search for the maximum entropy of the optimal threshold.

Region based algorithm includes the region growing method and the region splitting and merging method. The region growing method selects the initial seeds from the target region firstly, and judges whether the surrounding pixels could be merged into the growing area based on the regional growth criterion. The selection of initial points and the regional growth criterion are crucial. Region splitting and merging is usually based on the quadtree data theory, which would divide an image into a set of arbitrary, disconnected regions, then merge or split these regions to meet the segmentation conditions. Chen and Han [8] improved the image object segmentation result using the seeded region growing and k-means.

Edge based algorithm segments image by detecting edges or pixels with sharp variation in intensity between different regions, which are extracted and linked to form the boundary of the target. It is the balance between accuracy and interference that affects whether a satisfactory segmentation could be obtained. If the detection accuracy is too high, the noise would produce the false edge, which makes the image contour unreasonable; otherwise, if the noise immunity is too high, some parts of the image contour would be missed and the position of the target may be misjudged. Pan *et al.* [9] proposed a novel method of cell image segmentation by using the bacterial foraging based on edge detection.

Fuzzy set theory based on fuzzy membership mainly solves the uncertainties that are incomplete and ambiguous. Fuzzy set theory could describe the fuzziness and randomness of human vision preferably, thus, it is the methods using the fuzzy set theory that are widely used in many methods of image segmentation. Zheng *et al.* [10] proposed an adaptive image segmentation method that was based on the fuzzy c-means with the spatial information, the results showed that the method could segment the images with inhomogeneity and get better regional overlap measurement compared with some segmentation methods.

Artificial neural network is a computer information processing system that mimics the structure and function of brain. Up to now, many kinds of neural networks have been put forward such as back propagation neural network (BP) [11], radial basis function neural network [12], pulse coupled neural network (PCNN) [13], [14] and so on. Compared to traditional neural network, the PCNN does not need to learn, which has the biological background and is widely used in image processing, such as image segmentation [15], image fusion [16], feature extraction [17], and more applications of PCNN could be found from literatures [14].

In order to improve the performance of PCNN in image segmentation, many scholars have done lots of research. Zhou and Shao [18] proposed an extended PCNN model based on the strategy of the decision tree, and established the relationship between parameters and image features. He *et al.* [19] made the normalized spectral residual saliency as the linking coefficient and used the improved dynamic threshold based on the average gray values of the iteration segmentation to simplify the PCNN model. Lian *et al.* [20] used an optimal histogram threshold to determine the parameters of SPCNN for various images and added an offset to improve the segmentation precision. At the same time, some intelligent optimization methods are used to determine the parameters of PCNN for image segmentation. Zhang *et al.* [21] used the immune algorithm to determine the parameters of PCNN for the image segmentation of cotton leaf. Xu *et al.* [22] proposed the self-adaptive PCNN based on the ant colony optimization for the medical image segmentation, the image entropy was used as the fitness function. Hage and Hamade [23] used the PSO to determine the parameters of PCNN for segmentation of histology slides of cortical, the fitness function was entropy and energy of the bone micro-constituents.

Gravitational search algorithm (GSA) is a new stochastic optimization algorithm [24], which has been applied to image processing [25], clustering [26] control engineering [27] and so on. Mirjalili proposed the hybrid particle swarm algorithm and GSA to train the feedforward neural networks [28]. Sun and Zhang [29] proposed the hybrid genetic algorithm and gravitational search algorithm for the image segmentation that used the multilevel threshold. Gao *et al.* [30] combined GSA with chaos for the unconstrained numerical optimization. In order to improve the exploration and exploitation of GSA, we combined GSA with gbest agent memory ability, which was called GLGSA.

The research on PCNN is basically focused on the structure and parameters, especially the parameters, which determine the performance of PCNN. In order to further improve performance of PCNN for image segmentation, GLGSA was used to optimize the parameters of the simplified PCNN and the comprehensive indexes were used as the fitness function to improve the accuracy of segmentation.

The rest of the paper is organized as follows: The SPCNN model and the principles of PSO and GSA are presented in section II. The GLGSA is proposed and the performance of GLGSA is analyzed and compared with GSA and GGSA in section III. In section IV, the new fitness function is shown for image segmentation and the SPCNN was optimized by the GLGSA. Section V presents and discusses the experiment results. The conclusion is provided in the last section.

## II. RELATED WORK

In this section, the SPCNN is introduced according to the PCNN model, then the principles of PSO and GSA are explained.

### A. BASIC PCNN MODEL

The PCNN model is inspired by the explanation of the synchronous dynamics of neuronal activity in the visual cortex of the cat brain [14]. The fact that PCNN has only one layer and does not require any training that makes it different from the general artificial neural networks [31]; the typical network topology for PCNN is a two-dimensional planar lattice with a one-to-one pixel to neuron correspondence [32]. Just as shown in the Figure 1, a neuron of PCNN could be divided into three parts: input field, modulation field and

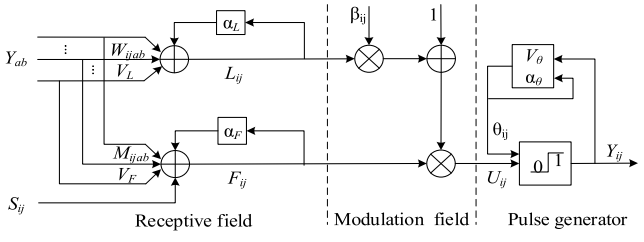


**FIGURE 1.** The model of standard PCNN.

pulse generator. The definition of PCNN mathematical model could be described as (1-5):

$$F_{ij}(n) = V_F \sum_{kl} M_{ijkl} Y_{kl}(n-1) + e^{-\alpha_F} F_{ij}(n-1) + S_{ij} \quad (1)$$

$$L_{ij}(n) = V_L \sum_{kl} W_{ijkl} Y_{kl}(n-1) + e^{-\alpha_L} L_{ij}(n-1) \quad (2)$$

where $F_{ij}$ and $L_{ij}$ denote the feedback input and the linking item, respectively. $M$ and $W$ are the weight matrices for the linking input. $V_F$ and $V_L$ represent the intrinsic electric potentials of $F_{ij}$ and $L_{ij}$, respectively. $S_{ij}$ is the input stimulus signal from the feeding field. $\alpha_F$ and $\alpha_L$ are the decay coefficients of $F_{ij}$ and $L_{ij}$, respectively.

$$U_{ij}(n) = F_{ij}(n)(1 + \beta L_{ij}(n)) \quad (3)$$

where $U_{ij}$ denotes the internal activity. $\beta$ represents the connection strength coefficient.

$$Y_{ij}(n) = \begin{cases} 1, & if\ \mathrm{U}_{ij}(n) > \theta_{ij}(n) \\ 0, & others \end{cases} \quad (4)$$

$$\theta_{ij}(n) = e^{-\alpha_\theta} \theta_{ij}(n-1) + V_\theta Y_{ij}(n-1) \quad (5)$$

here $Y_{ij}$ is the the pulse output. $\theta_{ij}$ is the dynamic threshold. The $V_\theta$ is the amplitude of dynamic threshold $\theta_{ij}$. $\alpha_\theta$ denotes the decay coefficient of $\theta_{ij}$.

The internal activity $U_{ij}$ is modulated by two inputs through the linking strength coefficient $\beta$; if the internal activity $U_{ij}$ is larger than the dynamic threshold $\theta_{ij}$, the neuron fires, which means that the pulse generator will be activated and impulse would be generated. Subsequently, the dynamic threshold would increase by the amplitude $V_\theta$ suddenly. If the internal activity is less than the dynamic threshold $\theta_{ij}$, the neuron could not fire and the dynamic threshold would decay by the factor $e_\theta^{-\alpha}$, the impulse generator would be closed, which means the output of neuron would equal to 0. When the internal activity $U_{ij}$ is larger than the dynamic threshold $\theta_{ij}$, the neuron would be fired again, the output of neuron would equal to 1. The process would constantly continue.

When the neuron generates a pulse, the pulse signal would be transmitted to the adjacent neurons, these neurons with the little gray value will automatically be fired quickly. Therefore, activation of a neuron would cause all the adjacent neurons with approximate gray value to be fired, which is the basic principle of image segmentation using the PCNN.

### B. SIMPLIFIED PCNN

So far, there have been lots of modified models, such as ICM [33], unit linking PCNN model [34], fast-linking PCNN model [35], SCM [36]. It has been proved that the SCM has lower computational complexity and higher accuracy rates for image segmentation when compared with other models [15]; so SPCNN which is derived from SCM model and showed in Figure 2 could be used in this paper and is described as followed:

$$L_{ij}(n) = \sum_{kl} M_{ijkl} Y_{kl}(n-1) \quad (6)$$

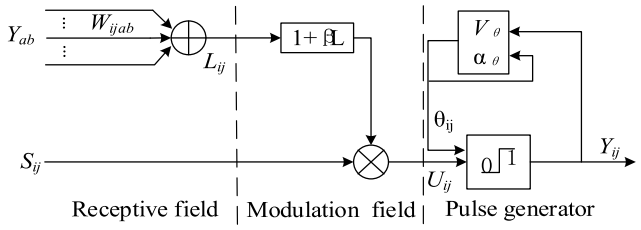


**FIGURE 2.** The model of SPCNN.

$$U_{ij}(n) = e^{-\alpha_f} U_{ij}(n-1) + S_{ij}(1 + \beta \sum_{kl} W_{ijkl} Y_{kl}(n-1)) \quad (7)$$

$$Y_{ij}(n) = \begin{cases} 1, & if\ \mathrm{U}_{ij}(n) > \theta_{ij}(n) \\ 0, & others \end{cases} \quad (8)$$

$$\theta_{ij}(n) = e^{-\alpha_\theta} \theta_{ij}(n-1) + V_\theta Y_{ij}(n-1) \quad (9)$$

where all the notations have the same meanings as indicated in (1-5). Compared with the SCM, the firing condition of SPCNN is $U_{ij}(n) > \theta_{ij}(n)$ rather than the sigmoid function in SCM. The principle of image segmentation using SPCNN is the same as that of PCNN.

There are still several adjustable parameters in the SPCNN model: $\alpha_f$, $\beta$, $\alpha_\theta$, $V_\theta$, and $n$. How to automatically set the values of these parameters would be explained in the fourth Section.

### C. THE PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

The PSO algorithm is also a stochastic, population-based optimization method, which is inspired by the social behavior of bird flocking or fishing schooling [37]. Suppose the following scenario: there is a group of birds looking for food in an area with only one piece of food. No bird knows the location of the food. But they know how far the current location is from the food. What is the best strategy for finding food? The easiest and most effective way is to follow the bird which is the nearest to the food. The PSO is analogous to this scenario and is used to solve optimization problems.

In PSO, each particle denotes a candidate solution to the optimization problem, which is adapted based on two attractors: the one is the particle has achieved iteratively so far, which is called pbest, the other is obtained by all particles in the population so far that is known as gbest and causes all particles to move towards the best particle. At each iteration, all the particles move at their own velocity and the movement is computed as followed:

$$v_i^{t+1} = w \times v_i^t + c_1 r_1 (pbest_i - x_i^t) + c_2 r_2 (gbest - x_i^t) \quad (10)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (11)$$

where $v_i^t$ is the velocity of particle $i$ at iteration $t$, $w$ is the inertia factor, $c_1$ and $c_2$ are the acceleration coefficient. $r_1$ and $r_2$ are the random number between 0 and 1. $x_i^t$ is the current position of $i$ at iteration $t$. pbest and gbest are the best position hunted by particle itself and the whole population so far, respectively, which reflects particle memory ability. The first

part of (10) $w \times v_i^t$ represents the exploration ability of the PSO algorithm, the second $c_1 r_1 (pbest_i\text{-}xt\ i)$ and third part $c_2 r_2 (gbest_i\text{-}xt\ i)$ denote the private thinking and collaboration of particles, respectively.

### D. THE GRAVITATIONAL SEARCH ALGORITHM (GSA)

The GSA algorithm is proposed by Rashedi (2009), which is inspired by the law of gravity and law of motion [24]. Each solution of the optimization problem is considered to be an agent $x_i$ which is running in an $n$-dimensional search space with the velocity $v_i$. The performance of the agents is weighed by their masses. The heavier masses that represent more efficient solutions move slower than the lighter ones in the exploitation process of algorithm, so the agents move toward those heavy agents, as illustrated in Figure 3.

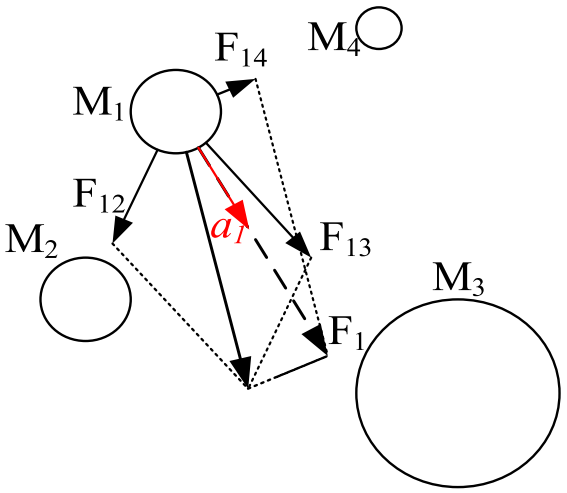


**FIGURE 3.** Each agent accelerates toward the total force that acts it from the other agents.

The position and velocity of each agent in the algorithm could be described respectively as:

$$X_i(t) = [x_i^1(t), x_i^2(t), \cdots, x_i^n(t)] i = 1, 2, \cdots, N$$
$$V_i(t) = [v_i^1(t), v_i^2(t), \cdots, v_i^n(t)] i = 1, 2, \cdots, N$$

**Law of gravity:** every agent attracts every other agent in the space with a special force which is proportional to the product of their masses and inversely proportional to the square of distance between them.

**Law of motion:** the current velocity of an object is equal to the sum of its previous velocity and acceleration. The acceleration of an object is directly proportional to the total farce and inversely proportional to the mass of the object.

The mass of agent could be calculated based on the current fitness of agent as followed:

$$m_i(t) = \frac{fit_i(t) - f_{worst}(t)}{f_{best}(t) - f_{worst}(t)} \tag{12}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{13}$$

where $M_i(t)$ denotes the mass; $fit_i(t)$ is the fitness value of the $i$th agent at the $t$th iteration; for a minimization problem, $f_{worst}(t)$ and $f_{best}(t)$ are the worst fitness value and the best fitness value at the $t$th iteration, respectively, which are given as followed:

$$f_{best}(t) = \min_{i \in \{1,2,\cdots,N\}} fit_i(t) \tag{14}$$

$$f_{worst}(t) = \max_{i \in \{1,2,\cdots,N\}} fit_i(t) \tag{15}$$

In the GSA algorithm, the mass of an agent could be defined conceptually as either passive gravitational mass $M_{pi}(t)$, active gravitational mass $M_{ai}(t)$, or the inertial mass $M_{ii}(t)$. The gravitational mass and inertial mass could be calculated using (16).

$$M_{pi}(t) = M_{ai}(t) = M_{ii}(t) = M_i(t) \tag{16}$$

According to the law of gravity, the gravitational force between the agent $i$ and agent $j$ could be calculated using (17).

$$F_{ij}^d(t) = G(t) \times \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon} \times (x_j^d(t) - x_i^d(t)) \tag{17}$$

where the $M_{pi}(t)$ and $M_{aj}(t)$ are the passive gravitational mass and active gravitational mass corresponding to the $i$th and $j$th agent, respectively. The $R_{ij}(t)$ denotes the Euclidian distance between the agent $i$ and $j$, the $\varepsilon$ is a small constant, which is used to prevent the denominator from being 0. The $G(t)$ represents the gravitational constant and could be defined as:

$$G(t) = G_0 \times \exp(-\alpha \times \frac{t}{T}) \tag{18}$$

where $G_0$ is the initial value of gravitational constant, the $\alpha$ stands for the descending coefficient. $T$ represents the maximum number of iteration.

Based on the law of motion, the acceleration of the agent $i$ at the $t$th iteration in the dimension $d$ is described by (19):

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{19}$$

$$F_i^d(t) = \sum_{j \in K_{best}, j \neq i}^{N} randf \times F_{ij}^d(t) \tag{20}$$

where $F_i^d(t)$ is the total force acting on the agent $i$ in the direction of dimension $d$ at $t$th iteration. $randf$ is a random variable between 0 and 1. $K_{best}$ is the set of $K$ number of best agents, which decreases linearly to 1 with initial value $N$. The agents require social interaction between them initially, which needs more agents. But the algorithm needs to search the best agents, so only a few agents are needed. Moreover, reduction of $K_{best}$ also reduces the computational complexity.

Finally, the velocity and position of each agent are updated based on the following (21-22). The flow chart of GSA is shown in Figure 4.

$$v_i^d(t+1) = randv \times v_i^d(t) + a_i^d(t) \tag{21}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{22}$$

where $v_d^i$ and $x_i^d$ stand for the velocity and position of the agent at the $d$th dimension, respectively.

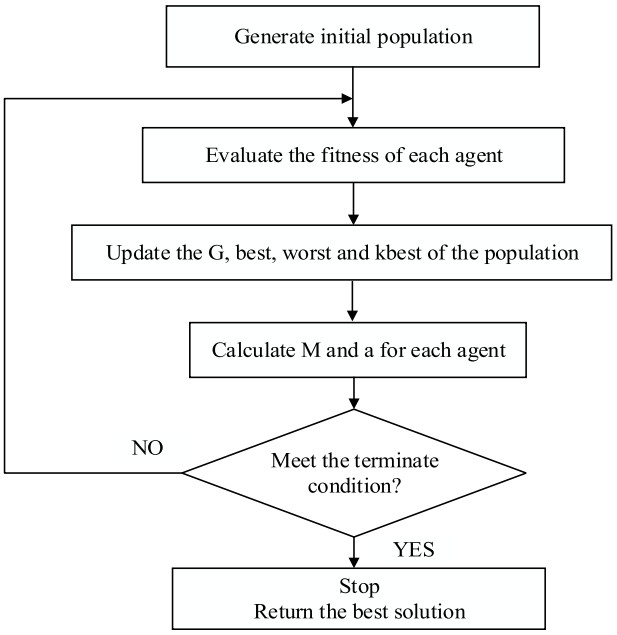


**FIGURE 4.** The flowchart of GSA algorithm.

## III. THE PROPOSED GBEST LED GRAVITATIONAL SEARCH ALGORITHM

In this section, we illustrate the disadvantage of GSA firstly, then propose our algorithm GLGSA and give the pseudo code. Finally, we compare it with GSA and GGSA, and analyze the results.

### A. THE SHORTCOMING OF GSA ALGORITHM

As discussed early, it is the exploration and exploitation of the algorithm that are main characteristics for all the population-based algorithm with evolutionary behavior. GSA and PSO are both optimization algorithms with the agents moving in the search space, but the strategies they use are different. The direction of agent in the GSA is calculated using the overall force obtained by the other agents, however, GSA has no memory ability for saving the position of the best agent in the population, so the best agent could be attracted away by the other particles of less masses and the best solution may be lost. PSO has the memory function and uses the social information among the agents, where the individuals could move towards the best position as fast as possible. Therefore we could save the position of the best agent to enhance the performance of GSA algorithm, which could be considered that the combination of GSA and gbest agent memory ability.

### B. THE GBEST LED GRAVITATIONAL SEARCH ALGORITHM

The proposed gbest led gravitational algorithm (GLGSA) has the innovative moving strategy, which has memory ability, shares the social information and is led by the social information gbest. As we know, most of agents become more efficient solutions after several iterations, which indicates that the mass would increase and the acceleration of agent would decrease relatively, the velocity change $a_1 \times d_t$ is very small in the $d_t$ time. In order to increase the velocity of the agent moving toward the optimal solution, we add velocity $v_g$ to the agent based the distance between agent and the best agent gbest, which could also be considered to be the leading role of gbest, just as shown in Figure 5.

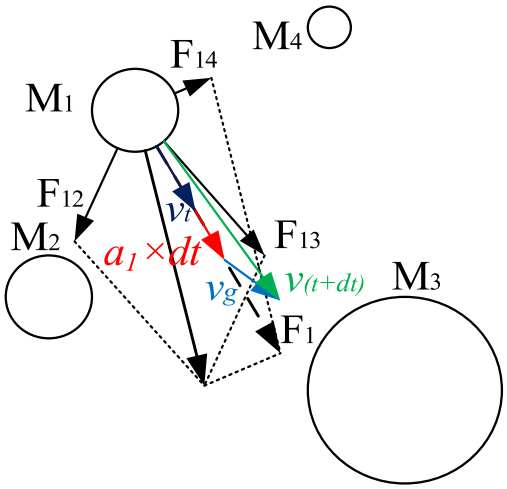


**FIGURE 5.** The velocity of agent is affected by the gbest.

In order to increase exploration and exploitation of algorithm, we use the linear method to combine the acceleration $\alpha_i^d(t)$ and the social information $(gbest - x_i^d(t))$ due to the change of mass and acceleration, and enhance the ability of agent memory to compensate for the decrease in acceleration. The velocity of the agent could be modified as:

$$v_i^d(t+1) = randv \times v_i^d(t) + c_1(t) \times a_i^d(t) + c_2(t) \times (gbest - x_i^d(t)) \quad (23)$$

$$c_1(t) = \lambda - \frac{t}{\max iter} \quad (24)$$

$$c_2(t) = \lambda - c_1(t) = \frac{t}{\max iter} \quad (25)$$

where $randv$ is a random variable in the rang [0,1], $\lambda$ is a constant, $c_1$ is the coefficient of the acceleration and $c_2$ is coefficient of the social information, just as shown in the Figure 6. $a_i^d$ is the acceleration of agent $i$ in the $d$th dimension. max$iter$ is the maximum iterations. The pseudo code of GLGSA is shown in Table 1.

It is significant to analyze the complexity of the algorithm. When only one main loop is considered, the time complexity of an algorithm is proportional to the number of the particles $N$ and the number of dimensions $D$. The computational step analysis for GSA, GGSA and GLGSA is $3ND+2N$ in an iterative loop. So the computational complexity of these algorithm is $O(ND)$.

### C. THE PERFORMANCE OF GLGSA

In order to fully prove the performance of our proposed algorithm GLGSA, 23 standard benchmark functions are employed in this section [38], which are represented

**TABLE 1.** The pseudo code of Gbest Led gravitational search algorithm (GLGSA).

| **Gbest Led Gravitational Search Algorithm (GLGSA):** |
|---|
| **Initialize** |
| Step 1: Set the parameters: $x_{min}$, $x_{max}$, N, Dim, ε, α, $G_0$, max_iter |
| Step 2: t=1 % t is the number of iteration % |
| Step 3: Random initialize the population position $X_i(1) \in X$ and velocity $V_i(1)$ |
| Step 4: Evaluate the fitness value of initial particles $fit_i(1)$ |
| Step 5: Get the global best position $gbest(1)$ |
| **Optimize** |
| Step 6: while $t \leq t_{max}$ do |
| Step 7: t=t+1 |
| Step 8: Calculate the gravitational constant G(*t*) by using (18) |
| Step 9: Calculate the coefficient of the acceleration and the social information $c_1$ and $c_2$ by using (24,25), respectively |
| Step 10: Update $f_{best}(t)$ and $f_{worst}(t)$ by using (14,15), respectively |
| Step 11: For $i = 1 : N$ |
| Step 12: Calculate the $M_i(t)$ by using (13) |
| Step 13: End for |
| Step 14: For $i = 1 : N$ |
| Step 15: For $d = 1 : Dim$ |
| Step 16: Calculate the acceleration $a_i^d(t)$ by using (19) |
| Step 17: Update the velocity $v_i^d(t)$ and position $x_i^d(t)$ by using (23, 22), respectively |
| Step 18: End for |
| Step 19: End for |
| Step 20: For i = 1 : N |
| Step 21: Evaluate the fitness value of each agent $fit_i(t)$ |
| Step 22: End for |
| Step 23: Update $gbest_i(t)$ |
| Step 24: End while |
| **Return results** |
| **Terminate** |

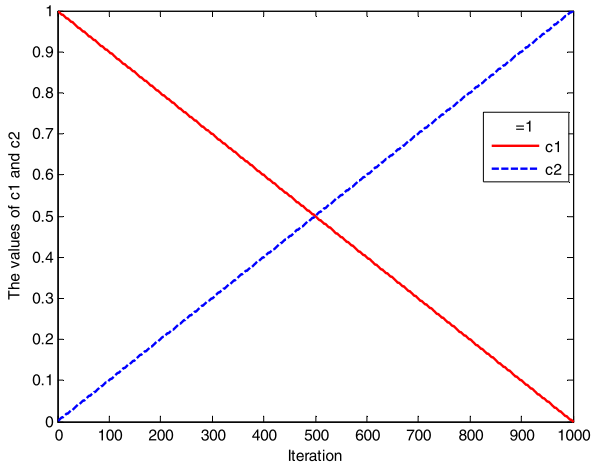


**FIGURE 6.** The values of the coefficients c1 and c2.

in Table 2–4 and could be divided into unimodal test functions, multimodal test functions and multimodal test functions with fixed dimension. GLGSA is compared with the GSA [24] and GGSA [39], whose parameters are shown in Table 5. The algorithms were run 30 times for each benchmark function, starting from the different randomly generated population. The best_so_far, average value and standard deviation are the best result, the mean value and standard deviation of the algorithm in 30 runs, respectively. The best_so_far, the average value and the standard deviation of the results are shown in Table 6–8, the convergence curve presented in Figure 7. In theory, the minimum value $f_{opt}$ of the functions and the corresponding $X$ values are shown in Table 9. $n$ denotes the dimension of function, $f_{opt}$ represents the minimum value of the function, $S$ is the subset of $R^n$.

### 1) UNIMODAL TEST FUNCTIONS

Functions $F_1$-$_7$ are the unimodal functions, which have only one global optimum and could be used to evaluate the exploitation capability of the algorithm, so the convergence rate of algorithm is more significant for the unimodal functions. It is obvious that the proposed GLGSA algorithm is competitive with GSA and GGSA from the Table 6. The function $F_7$ is a quartic function with a random item, so it is difficult to find the global optimal solution. The convergence rate of GLGSA performs better than the other two

**TABLE 2.** Unimodal test functions.

| *Test function* | *S* |
|---|---|
| $F_1(X)=\sum_{i=1}^{n}x_i^2$ | $[-100,100]^n$ |
| $F_2(X)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$ | $[-10,10]^n$ |
| $F_3(X)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2$ | $[-100,100]^n$ |
| $F_4(X)=\max\{\lvert x_i\rvert,1\le i\le n\}$ | $[-100,100]^n$ |
| $F_5(X)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$ | $[-100,100]^n$ |
| $F_6(X)=\sum_{i=1}^{n}([x_i+0.5])^2$ | $[-100,100]^n$ |
| $F_7(X)=\sum_{i=1}^{n}ix_i^4+random[0,1)$ | $[-100,100]^n$ |

**TABLE 3.** Multimodal test functions.

| *Test function* | *S* |
|---|---|
| $F_8(X)=\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$ | $[-500,500]^n$ |
| $F_9(X)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$ | $[-5.12,5.12]^n$ |
| $F_{10}(X)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$ | $[-32,32]^n$ |
| $F_{11}(X)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2+\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$ | $[-600,600]^n$ |
| $F_{12}(X)=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i=1}^{n}u(x_i,10,100,4)$<br>$y_i=1+\frac{x_i+1}{4}$<br>$u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)^m & x_i<-a\end{cases}$ | $[-50,50]^n$ |
| $F_{13}(\mathrm{X})=0.1\{\sin^2(3\pi\,\mathrm{x}_1)+\sum_{i=1}^{n}(\mathrm{x}_i-1)^2[1+\sin^2(3\pi\,\mathrm{x}_i)+1]+(\mathrm{x}_n-1)^2[1+\sin^2(2\pi\,\mathrm{x}_n)]\}+\sum_{i=1}^{n}u(\mathrm{x}_i,5,100,4)$ | $[-50,50]^n$ |

algorithm which could be seen from the Figure 7 (F1-F7), therefore, we could say that GLGSA has the higher convergence rate.

### 2) MULTIMODAL TEST FUNCTIONS

Functions $F_{8\text{-}13}$ are high dimensions and have large numbers of local optimum, which are more difficult to optimize than the unimodal functions. If the algorithm could escape from the all local optimum and converge to the global optimum, the algorithm has better exploration. The best_so_far and average in Table 7 evidence that the proposed algorithm GLGSA has better exploration capability. The standard deviation results of $F_8$ and $F_9$ indicate that the results of GLGSA fluctuate greatly sometimes. But the standard deviation results of $F_{10\text{-}13}$ show that the stability of GLGSA is better GGGSA. And it could be observed that convergence rate of GLGSA algorithm is far better than GSA and GGSA from the Figure 7 (F8-F23). Therefore, it could be stated that the GLGSA algorithm could explore the more promising regions of space and has better exploration ability, which could avoid the local optimum and converge to the global optima with very large probability.

### 3) MULTIMODAL TEST FUNCTIONS WITH FIXED DIMENSION

Functions $F_{14\text{-}23}$ are low dimensions, which have a lower number of local optimum than the multimodal test functions.

**TABLE 4.** Multimodal test functions with fixed dimension.

| *Test function* | *S* |
|---|---|
| $F_{14}(X)=(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6})^{-1}$ | $[-65.53,65.63]^2$ |
| $F_{15}(X)=\sum_{i=1}^{11}[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$ | $[-5,5]^4$ |
| $F_{16}(X)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+x_2^4$ | $[-5,5]^2$ |
| $F_{17}(X)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$ | $[-5,10]\times[0,15]$ |
| $F_{18}(X)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times$ $[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$ | $[-5,5]^2$ |
| $F_{19}(X)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$ | $[0,1]^3$ |
| $F_{20}(X)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$ | $[0,1]^6$ |
| $F_{21}(X)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$ | $[0,10]^4$ |
| $F_{22}(X)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$ | $[0,10]^4$ |
| $F_{23}(X)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$ | $[0,10]^4$ |

**TABLE 5.** Initial parameters.

| Parameter | GSA | GGSA | GLGSA |
|---|---|---|---|
| Number of agents | 30 | 30 | 30 |
| Dimension | 30 | 30 | 30 |
| $G_0$ | 100 | 100 | 100 |
| α | 20 | 20 | 20 |
| Max iteration | 1000 | 1000 | 1000 |
| Terminating conditions | Max iteration | Max iteration | Max iteration |
| $C_1$ | / | $(-2t^3/T^3)+2$ | $(-t/T)+\lambda$ |
| $C_2$ | / | $2t^3/T^3$ | $\lambda-C_1$ |

We could apply these functions to test the robustness of the algorithm. The best_so_far and average of Functions in Table 8 prove that GLGSA has better robustness than the other two algorithms except $F_{16}$, $F_{17}$ and $F_{18}$. The three algorithms perform the same for the function $F_{16}$, $F_{17}$ and $F_{18}$, but from the perspective of standard deviation, the stability of GSA algorithm shows better for the function $F_{16}$ and the GLGSA does better for the function $F_{18}$. Although the stability of algorithm GLGSA is not the best for the function $F_{20}$ and $F_{22}$, from the view of the best_so_far and average, the GLGSA does the best. Finally the convergence curves in Figure 7 (F14-F23) make it known that the GLGSA could converge on the global optima.

In general, these results indicate that the GLGSA has better ability of convergent and avoidance local optimal solution, which implies that GLGSA could properly balances the exploitation and exploration in the process of optimization.

## IV. SIMPLIFIED PCNN OPTIMIZED BY GBEST LED GRAVITATIONAL SEARCH ALGORITHM FOR IMAGE SEGMENTATION

The fitness function is explained firstly, then we introduce our proposed image segmentation algorithm carefully in this section.

### A. FITNESS FUNCTION

The fitness function is the most important factor for the swarm optimization algorithm, which evaluates the quality of agents. In most papers that use swarm optimization methods to determine the parameters of PCNN [21]–[23], it is

**TABLE 6.** The minimization results of unimodal test functions.

| Function | Parameter | GSA | GGSA | GLGSA |
|---|---|---|---|---|
| F1 | Best_so_far | 6.0759e-017 | 7.8973e-019 | 1.2175e-019 |
| | Average | 1.2317e-016 | 6.2683e-018 | 5.4456e-019 |
| | Standard deviation | 9.8634e-017 | 4.4781e-018 | 4.7569e-019 |
| F2 | Best_so_far | 3.3229e-008 | 5.1215e-009 | 1.7956e-009 |
| | Average | 6.0005e-008 | 1.3059e-008 | 4.6800e-009 |
| | Standard deviation | 2.9556e-008 | 6.6640e-009 | 4.9611e-009 |
| F3 | Best_so_far | 1.9748e+002 | 1.6206e+002 | 9.8458e+001 |
| | Average | 5.3312e+002 | 4.6152e+002 | 2.0917e+002 |
| | Standard deviation | 1.6012e+002 | 1.5965e+002 | 1.0022e+002 |
| F4 | Best_so_far | 6.9827e-001 | 1.0142e-008 | 7.7568e-009 |
| | Average | 3.3097e+000 | 1.8086e-008 | 1.2003e-008 |
| | Standard deviation | 4.2434e+000 | 1.3226e-008 | 6.3027e-009 |
| F5 | Best_so_far | 2.4376e+001 | 2.5637e+001 | 2.4265e+001 |
| | Average | 3.8834e+001 | 5.0339e+001 | 3.0854e+001 |
| | Standard deviation | 4.8027e+001 | 8.4994e+001 | 2.1416e+001 |
| F6 | Best_so_far | 0 | 0 | 0 |
| | Average | 2.3333e-001 | 1.6667e-001 | 6.6667e-002 |
| | Standard deviation | 9.3526e-001 | 3.7905e-001 | 2.5371e-001 |
| F7 | Best_so_far | 1.9127e-002 | 2.0280e-002 | 1.3683e-002 |
| | Average | 1.9194e-001 | 1.0771e-001 | 7.3302e-002 |
| | Standard deviation | 4.8898e-001 | 9.6731e-002 | 1.8111e-001 |

**TABLE 7.** The minimization results of multimodal test functions.

| function | parameter | GSA | GGSA | GLGSA |
|---|---|---|---|---|
| F8 | Best_so_far | -3.2656e+003 | -3.6367e+003 | -7.6185e+003 |
| | Average | -2.6660e+003 | -2.7160e+003 | -5.9897e+003 |
| | Standard deviation | 6.4242e+002 | 4.2359e+002 | 6.9575e+002 |
| F9 | Best_so_far | 3.2834e+001 | 3.8803e+001 | 2.3879e+001 |
| | Average | 5.1738e+001 | 4.4441e+001 | 3.3829e+001 |
| | Standard deviation | 2.3185e+001 | 6.6247e+000 | 8.8434e+000 |
| F10 | Best_so_far | 5.8873e-009 | 9.1141e-010 | 2.0509e-010 |
| | Average | 8.0075e-009 | 1.9439e-009 | 5.9638e-010 |
| | Standard deviation | 1.5670e-009 | 9.6276e-010 | 2.7938e-010 |
| F11 | Best_so_far | 2.9560e+000 | 0 | 0 |
| | Average | 8.3529e+000 | 1.5826e-001 | 3.1101e-002 |
| | Standard deviation | 3.4362e+000 | 2.8154e-001 | 3.9607e-002 |
| F12 | Best_so_far | 3.1713e-019 | 1.3225e-020 | 6.9223e-022 |
| | Average | 1.2294e-001 | 8.4179e-002 | 5.5828e-002 |
| | Standard deviation | 2.7658e-001 | 1.3962e-001 | 1.3871e-001 |
| F13 | Best_so_far | 5.1136e-018 | 1.4385e-019 | 7.8173e-021 |
| | Average | 5.5206e-002 | 4.2854e-002 | 1.6732e-002 |
| | Standard deviation | 1.8197e-001 | 1.3344e-001 | 6.7170e-002 |

information entropy, mutual information or cross entropy that is used as the fitness function. Under the condition that the background and target are segmented by modulation and neural suppression character of PCNN, a novel fitness function was applied for the sake of preserving more contour details and reducing the noise caused by false segmentation in the paper. The fitness function consisted of cross entropy parameter (CEP), edge matching (EM) and noise control (NC).

The cross entropy is usually used to quantify the difference between two probability distributions, which could be used to measure the difference of target and background probability distributions between original image and segmented image. The smaller the cross entropy, the smaller the

**TABLE 8.** The minimization results of multimodal test functions with fixed dimension.

| function | parameter | GSA | GGSA | GLGSA |
|---|---|---|---|---|
| F14 | Best_so_far | 9.9800e-001 | 9.9800e-001 | 9.9800e-001 |
| | Average | 3.4002e+000 | 2.9017e+000 | 2.0862e+000 |
| | Standard deviation | 2.3242e+000 | 1.4852e+000 | 1.5987e+000 |
| F15 | Best_so_far | 1.8681e-003 | 1.7661e-003 | 5.4171e-004 |
| | Average | 5.2843e-003 | 4.7627e-003 | 1.6547e-003 |
| | Standard deviation | 3.5330e-003 | 4.0095e-003 | 3.5418e-003 |
| F16 | Best_so_far | -1.0316e+000 | -1.0316e+000 | -1.0316e+000 |
| | Average | -1.0316e+000 | -1.0316e+000 | -1.0316e+000 |
| | Standard deviation | 5.2964e-016 | 6.2532e-016 | 6.1849e-016 |
| F17 | Best_so_far | 3.9789e-001 | 3.9789e-001 | 3.9789e-001 |
| | Average | 3.9789e-001 | 3.9789e-001 | 3.9789e-001 |
| | Standard deviation | 0 | 0 | 0 |
| F18 | Best_so_far | 3.0000e+000 | 3.0000e+000 | 3.0000e+000 |
| | Average | 3.0000e+000 | 3.0000e+000 | 3.0000e+000 |
| | Standard deviation | 2.9527e-015 | 1.6880e-015 | 1.6431e-015 |
| F19 | Best_so_far | -3.8249e+000 | -3.8628e+000 | -3.8628e+000 |
| | Average | -3.2810e+000 | -3.8022e+000 | -3.8628e+000 |
| | Standard deviation | 3.8952e-001 | 3.2412e-001 | 2.5094e-015 |
| F20 | Best_so_far | -3.0314e+000 | -3.3220e+000 | -3.3220e+000 |
| | Average | -1.4710e+000 | -2.5380e+000 | -3.0352e+000 |
| | Standard deviation | 5.1752e-001 | 1.0004e+000 | 5.3020e-001 |
| F21 | Best_so_far | -5.0552e+000 | -5.0552e+000 | -5.0552e+000 |
| | Average | -4.0495e+000 | -3.4737e+000 | -4.7579e+000 |
| | Standard deviation | 1.7496e+000 | 2.1171e+000 | 9.4724e-001 |
| F22 | Best_so_far | -1.0403e+001 | -1.0403e+001 | -1.0403e+001 |
| | Average | -6.4797e+000 | -5.0243e+000 | -8.1376e+000 |
| | Standard deviation | 3.3343e+000 | 1.9363e+000 | 2.9179e+000 |
| F23 | Best_so_far | -1.0536e+001 | -1.0536e+001 | -1.0536e+001 |
| | Average | -8.8124e+000 | -1.0176e+001 | -1.0536e+001 |
| | Standard deviation | 3.0144e+000 | 1.3720e+000 | 1.4378e-015 |

**TABLE 9.** The optimal values of the functions in theory.

| *Test function* | $X_{opt}$ | $f_{opt}$ |
|---|---|---|
| $F_{1}$-$_{4}$ | $[0]^n$ | 0 |
| $F_5$ | $[1]^n$ | 0 |
| $F_6$ , $F_7$ | $[0]^n$ | 0 |
| $F_8$ | $[420.96]^n$ | -418.9829×$n$ |
| $F_{9}$-$_{11}$ | $[0]^n$ | 0 |
| $F_{12}$ | $[1]^n$ | 0 |
| $F_{13}$ | $[1]^n$ | 0 |
| $F_{14}$ | (-32,32) | 1 |
| $F_{15}$ | (0.1928,0.1908,0.1231,0.1358) | 0.00030 |
| $F_{16}$ | (0.089,-0.712),(-0.089,0.712) | -1.0316 |
| $F_{17}$ | (-3.14,12.27),(3.14,2.275),(9.42,2.42) | 0.398 |
| $F_{18}$ | (0,-1) | 3 |
| $F_{19}$ | (0.114,0.556,0.852) | -3.86 |
| $F_{20}$ | (0.201,0.15,0.477,0.275,0.311,0.657) | -3.32 |
| $F_{21}$ | 5 local minima in $a_i$ $i$= 1,⋯,5 | -10.1532 |
| $F_{22}$ | 7 local minima in $a_i$ $i$= 1, ⋯,7 | -10.4028 |
| $F_{23}$ | 10 local minima in $a_i$ $i$= 1, ⋯,10 | -10.5363 |

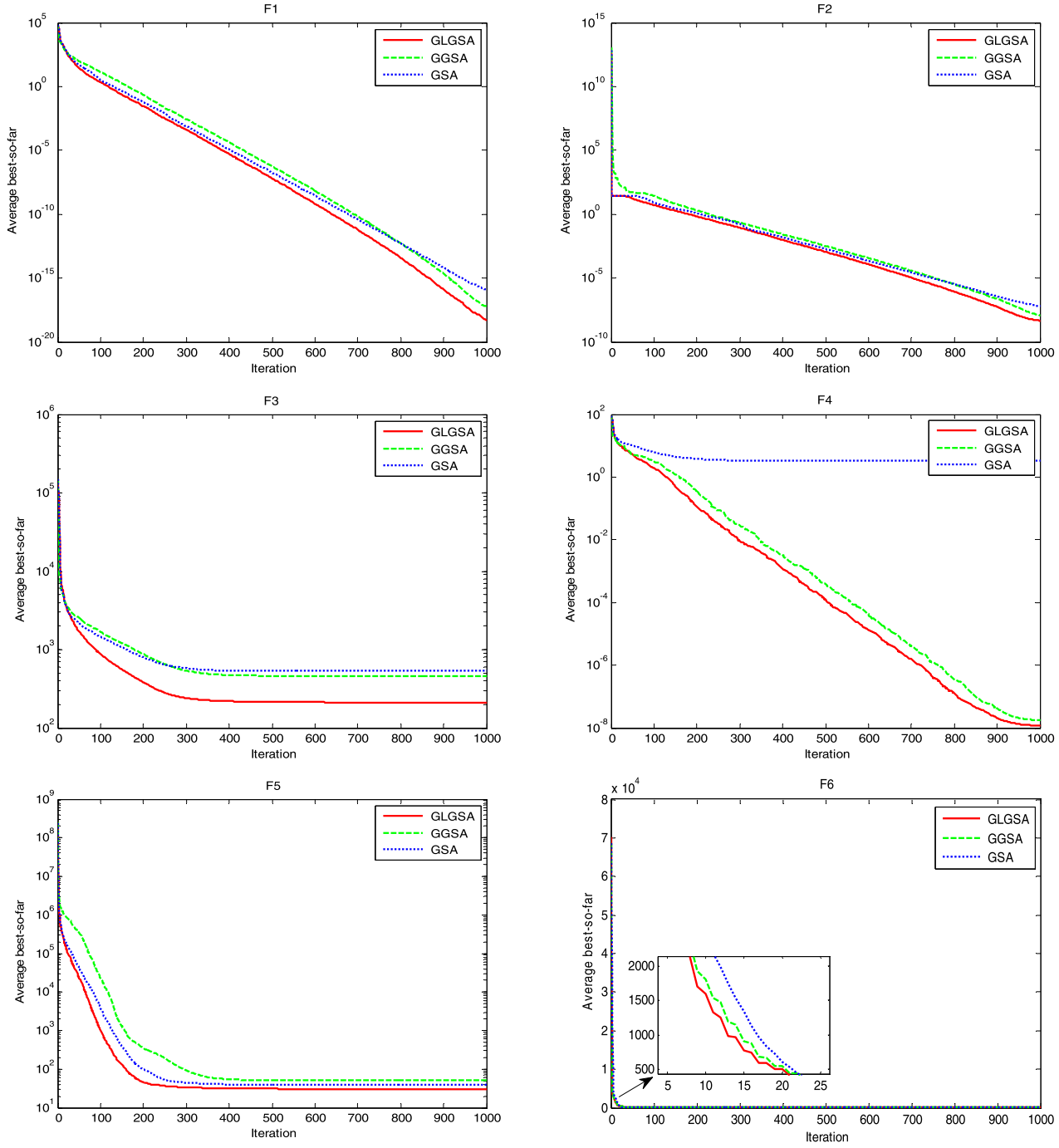


**FIGURE 7a.** The convergence graphs of GLGSA, GGSA and GSA on the benchmark functions.

difference between the target and background before and after segmented.

$$CE(P, Q : t) = \sum_{f=1}^{t}[f \times h(f) \times \ln \frac{f}{u_1(t)} + u_1(t) \times h(f) \times \ln \frac{u_1(t)}{f}] + \sum_{f=t+1}^{Z} [f \times h(f) \times \ln \frac{f}{u_2(t)} + u_2(t) \times h(f) \times \ln \frac{u_2(t)}{f}] \quad (26)$$

$$u_1(t) = \frac{1}{\sum_{f=0}^{t} h(f)} \sum_{f=0}^{t} f \times h(f) \quad (27)$$

$$u_2(t) = \frac{1}{\sum_{f=t+1}^{Z} h(f)} \sum_{f=t+1}^{Z} f \times h(f) \quad (28)$$

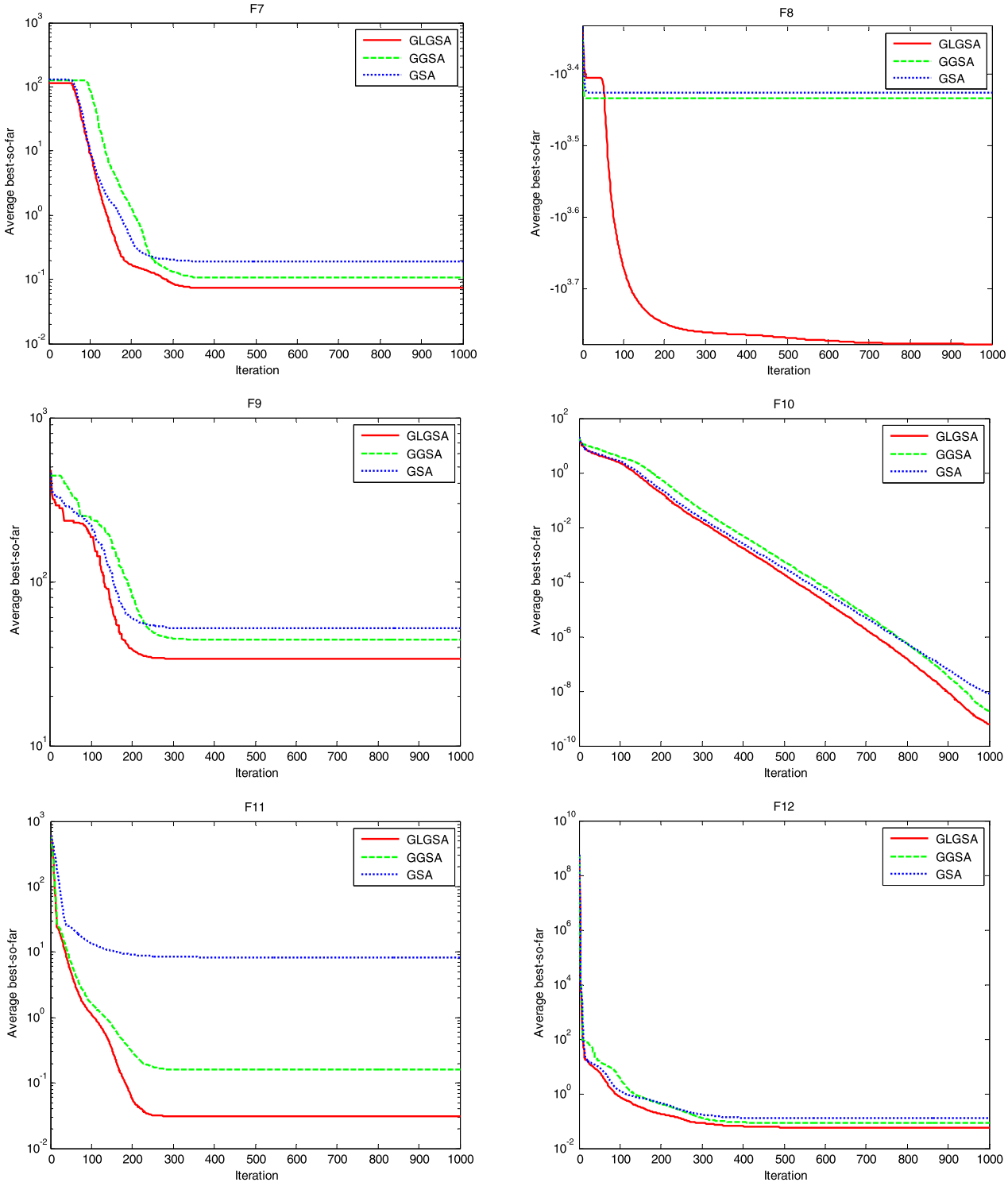


**FIGURE 7b.** ***(Continued.)*** **The convergence graphs of GLGSA, GGSA and GSA on the benchmark functions.**

where $CE$ is the cross entropy, $f$ is the gray value of the image, $h(f)$ denotes the histogram of image grayscale statistic, $Z$ represents the maximum gray value, $u_1$ and $u_2$ are the mean values within the class, representing the average gray value of the target and background, respectively.

$$CEP = 1 - \frac{CE}{Z} \tag{29}$$

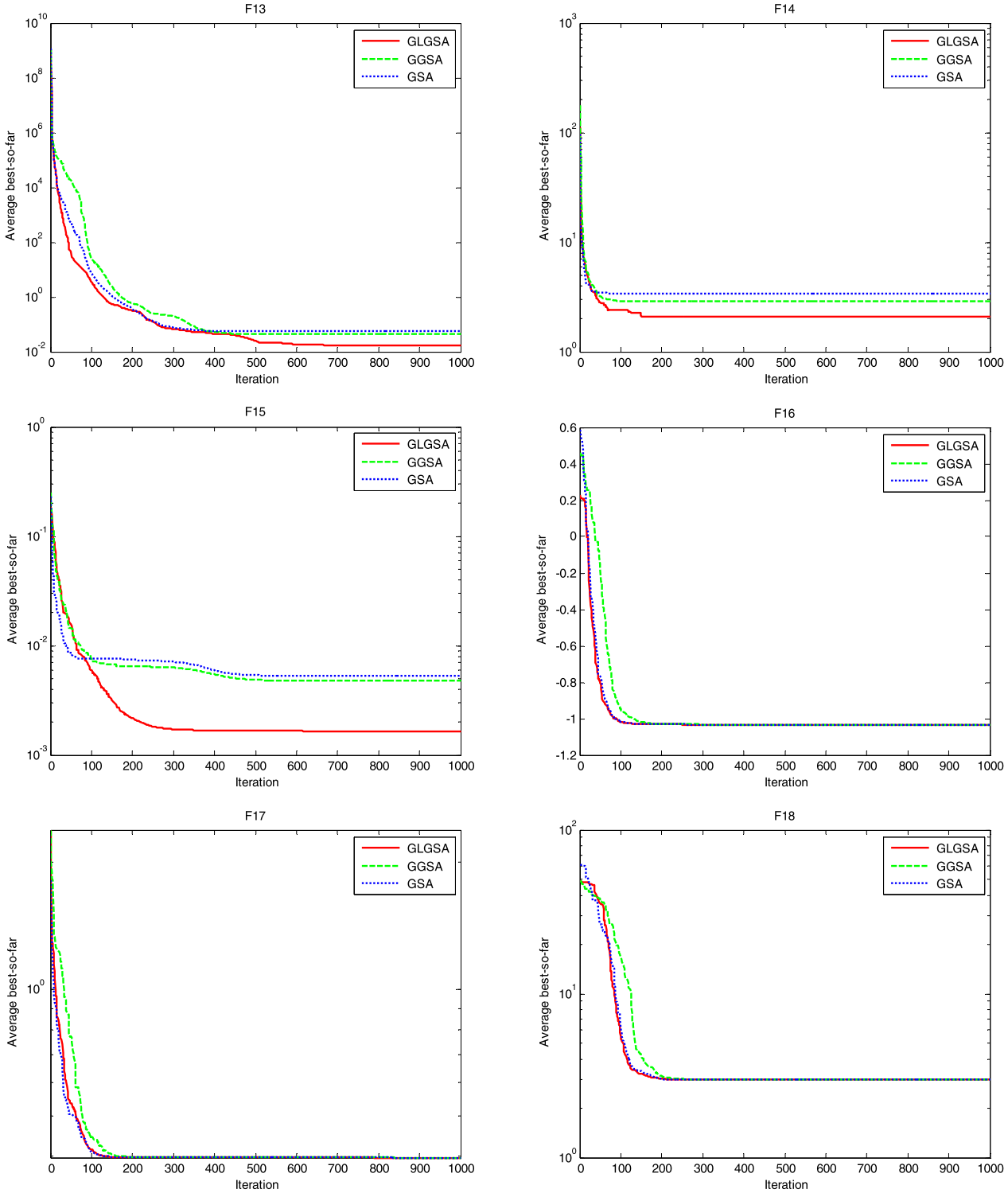


**FIGURE 7c.** ***(Continued.)*** **The convergence graphs of GLGSA, GGSA and GSA on the benchmark functions.**

The cross entropy parameter (CEP) is obtained based on normalization of cross entropy. The closer the CEP is to 1, the more accurate the segmentation is.

In the PCNN model, many neurons emit pulses at the same time because of the capture characteristics, but the pixels with the same gray value may be the target or the background in reality, thus how to segment these pixels correctly is important. The more texture details could be retained by using the edge guided neuron pixels to synchronize pulses. The Soble differential operator is applied for

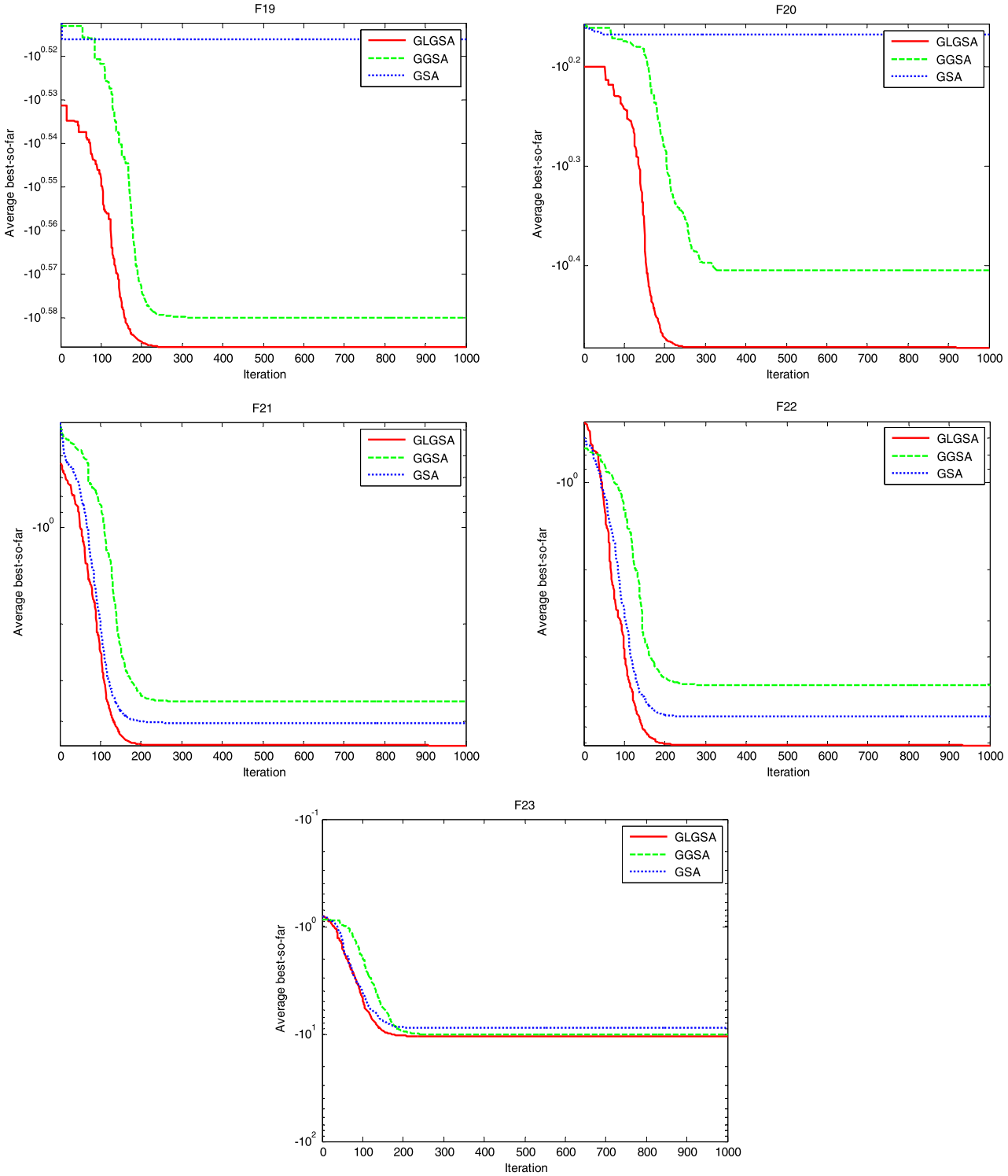


**FIGURE 7d.** ***(Continued.)*** **The convergence graphs of GLGSA, GGSA and GSA on the benchmark functions.**

edge detection.

$$EM = \frac{\sum (Yedge(i,j) \times Iedge(i,j))}{\sum Iedge(i,j)} \tag{30}$$

here *Iedge* and *Yedge* represent the edges of the original image and segmented image, respectively. The larger the *EM*, the richer the detail preservation of the segmented image.

In order to prevent the occurrence of noise due to wrong segmentation and preserve the details of the object contour,

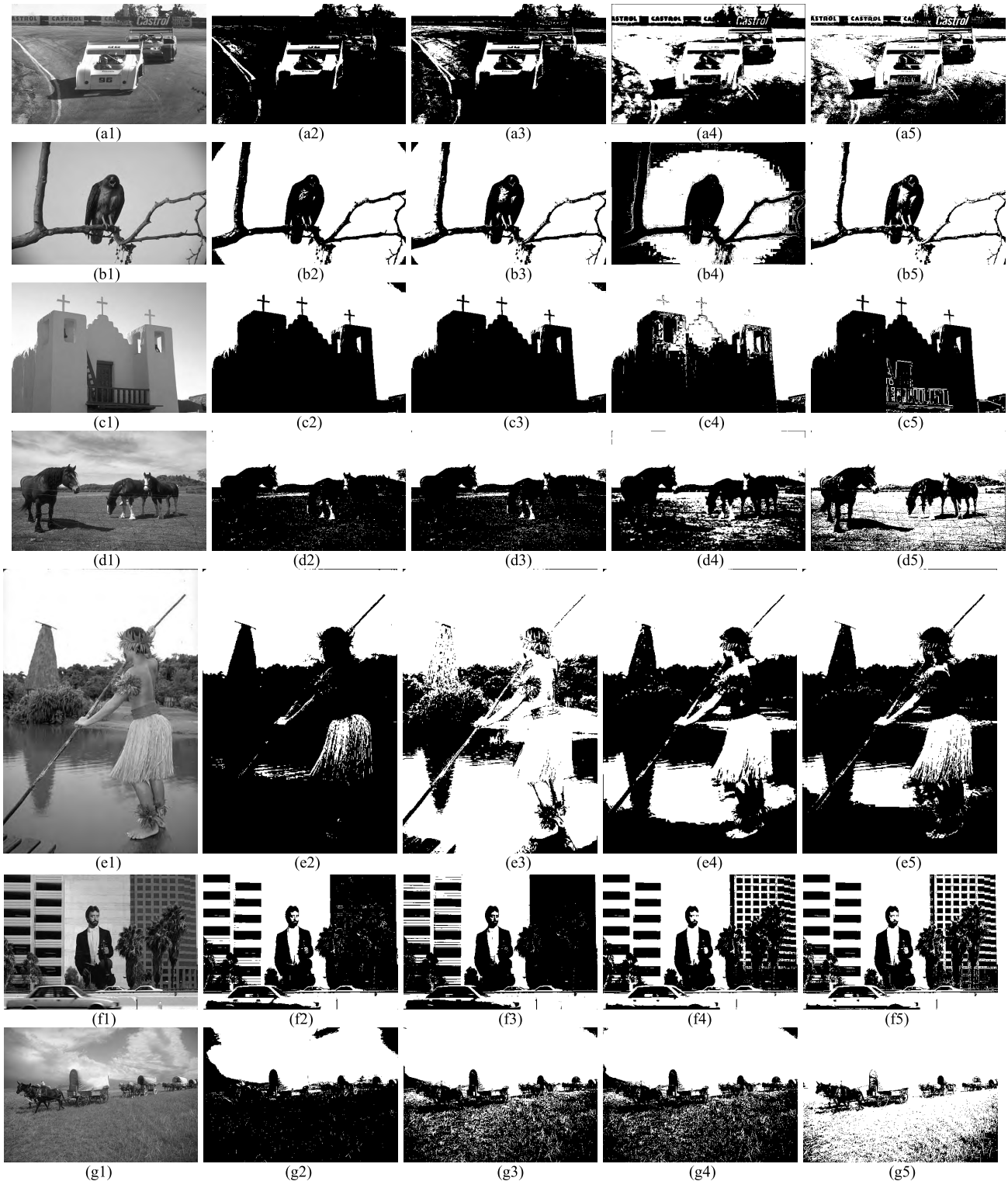


**FIGURE 8a.** The segmentation results of nine gray images.

the noise control is added to the fitness function. If there are no other target pixels in the 8-adjacent neighborhood, it is considered as discontinuous segmentation point, namely salt noise. Similarly, if there are no other background pixels in the 8-adjacent neighborhood, it is regarded as pepper noise.

$$V(i,j) = H \times Y(i,j) \tag{31}$$

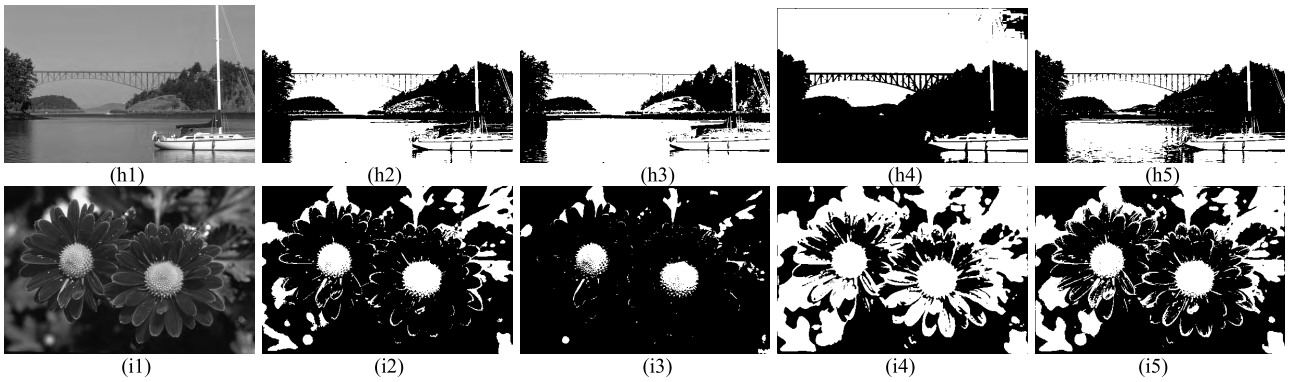


**FIGURE 8b.** *(Continued.)* **The segmentation results of nine gray images.**

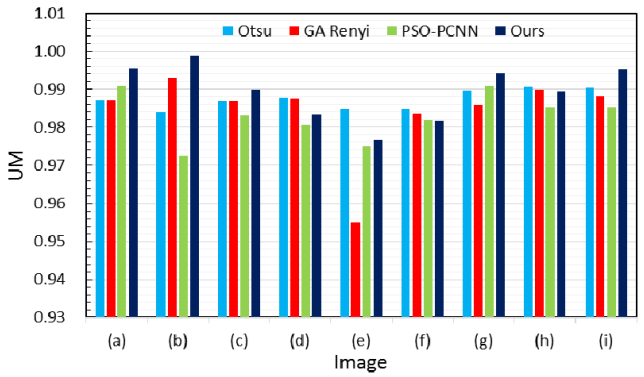


**FIGURE 9.** **The UM index of nine images segmented by four methods.**

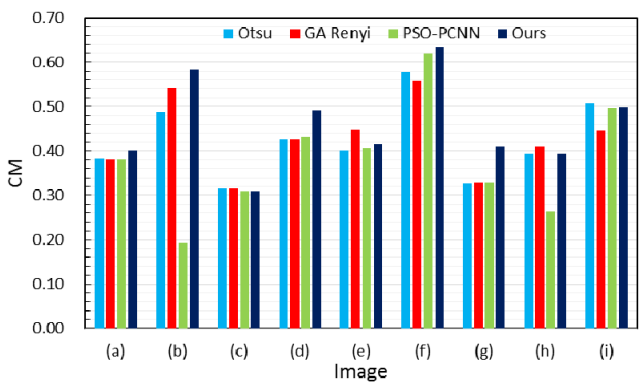


**FIGURE 10.** **The CM index of nine images segmented by four methods.**

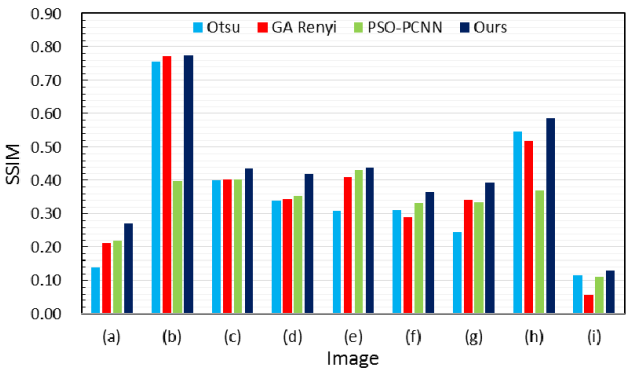


**FIGURE 11.** **The SSIM index of nine images segmented by four methods.**

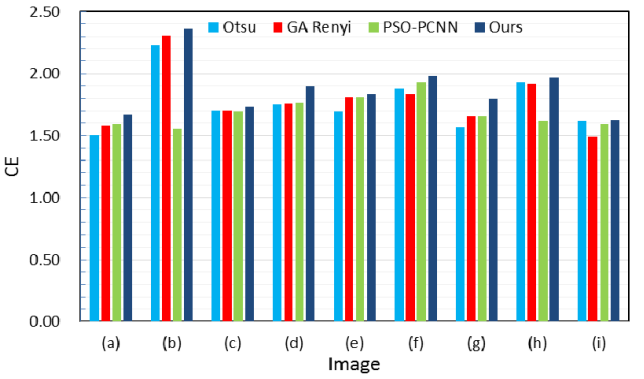


**FIGURE 12.** **The CE index of nine images segmented by four methods.**

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{32}$$

$$Saltnoise(i,j) = \begin{cases} 1, & Y(i,j) = 1, V(i,j) = 1 \\ 0, & others \end{cases} \tag{33}$$

$$Peppernoise(i,j) = \begin{cases} 1, & Y(i,j) = 0, V(i,j) = 8 \\ 0, & others \end{cases} \tag{34}$$

$$NC = 1 - \frac{\sum Saltnoise(i,j) + \sum Peppernoise(i,j)}{k \times N} \tag{35}$$

where $Y(i, j)$ is the number of pixels in the 8-adjacent neighborhood of pixel $Y(i,j)$, *Saltnoise* and *peppernoise* are the distribution marks of salt noise and pepper noise, respectively. $K$ is the normalized parameter, $N$ is the total number of pixels. The larger $NC$ is, which indicates that the segmented image is less affected by the noise.

In order to preserve contour and texture details while segmenting the target and background accurately, the fitness function could be described as:

$$fitness = \frac{1}{3}(CEP + EM + NC) \tag{36}$$

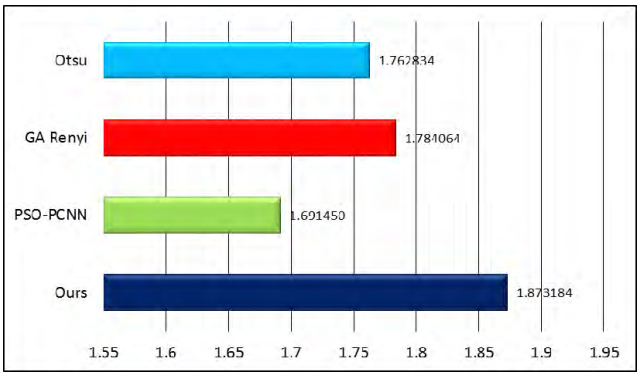


**FIGURE 13.** The average of CE index using four methods for all the images.

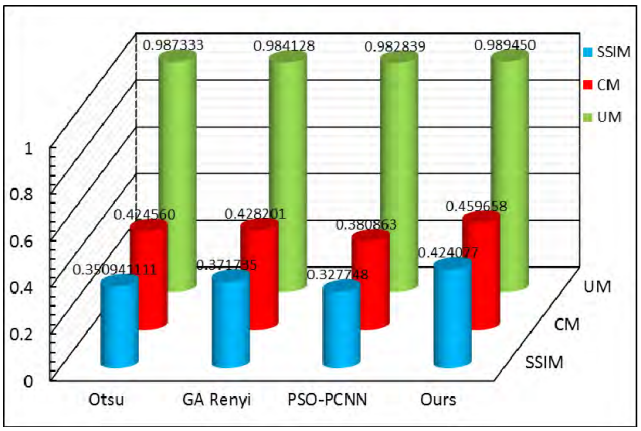


**FIGURE 14.** The three-dimensional graph of the final segmentation results: Data in the first, second and third raw represent the average values of UM, CM and SSIM corresponding to all images, respectively, which used four different methods.

### B. DESCRIPTION OF GLSGA-SPCNN ALGORITHM

**Step1**. Initialize algorithm parameters, the population size is set to 50 in this paper, there are four parameters to be optimized, which are $\alpha_f$, $\beta$, $\alpha_\theta$ and $V_\theta$. So the dimension of the agent is set to 4. Initialize all the agents position $(X_1^t, X_2^t, \ldots, X_N^t)$ in the search range $[X_{min}, X_{max}]^4$ randomly.

**Step2**. Evaluate the fitness (*fitt 1*,*fitt 2*,...,*fitt N*) of each agent in the population. The fitness is calculated by the fitness function based on SPCNN output, in order to improve the efficiency of the algorithm, let SPCNN iterate several times, take the maximum fitness as the fitness of the agent and output the corresponding the number $n$ of SPCNN iteration.

**Step3**. Update the *gravitational constant, gbest, best* and *worse* for the population based on the fitness values.

**Step4**. Calculate the inertial mass and external force of each agent for all agents.

**Step5**. Calculate the acceleration $a$, update the velocity $v$ and position $x$ for all agents.

**Step6**. Judge whether the termination conditions or achieve the maximum iterations is met. If not, return to **Step 2**. The termination condition is that the fitness value changes less than the preset value in three successive iteration.

## V. THE EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to assess the the effectiveness of the proposed image segmentation method, we compared it with the Otsu [1], GA Renyi [40] and PSO-PCNN [41], many images having specific feature were selected from the Berkeley Segmentation Dataset to do experiments [42]. Because of the limited space, we choose nine pictures for careful comparative analysis here, and then directly give some original images, corresponding segmented image and the parameters of SPCNN.

### A. DIRECT COMPARISONS BY THE SUBJECTIVE VISUAL PERCEPTION

Each row illustrates the segmentation results of an image using different methods which is shown in Figure 8. The images in the column (1) is the original images, the images in the column (2), (3), (4) and (5) denote segmentation results of the Otsu method, the GA Renyi method, the PSO-PCNN method and our proposed method, respectively. The parameters of PCNN using PSO optimization are shown in Table 10, the fitness values and parameters of SPCNN using GLGSA optimization are revealed in Table 11.

From the Figure 8, it is obvious that the segmented images using our proposed method are much better than the classical method Otsu, GA Renyi and PSO-PCNN. For the (a1), (d1), (e1) and (f1), our proposed method and PSO-PCNN have better performance, but it is the vehicle number in (a1), the fence wire and horse's texture in (d1), wrinkles of skirts in the (e1) and the branches and leaves of trees in the (f1) that are clearer compared our proposed method with PSO-PCNN. For the (b1) and (g1), our proposed method and GA Renyi method are more satisfactory, but the beak in the (b5) and the carriage in the (g5) are more visible. Our proposed method and Otsu method seem to be similar for (i1), but if we analyze them carefully, we would notice that the spots on the leaves are more evident in (i5) using our method.

### B. QUANTITATIVELY EVALUATE THE PERFORMANCE OF OUR PROPOSED METHOD

Besides the direct comparisons by the subjective visual perception, we quantitatively evaluate the performance of our proposed method using the uniformity measure (UM), region contrast measure (CM) [43], structural similarity (SSIM) [44] and comprehensive evaluation (CE), where CE is the sum of UM, CM and SSIM.

The UM is the uniformity of the segmented region which could be written as:

$$UM = 1 - \frac{1}{C}\sum_i \left\{ \sum_{(x,y)\in R_j} \left[ f(x,y) - \frac{1}{A_j} \sum_{(x,y)\in R_j}^{n} f(x,y) \right]^2 \right\} \tag{37}$$

**TABLE 10.** The parameters of pso-pcnn for the images in Figure 8.

| *Image* | $\alpha_\theta$ | $V_\theta$ | $\alpha_L$ | $V_L$ | $\alpha_F$ | $V_F$ | $\beta$ | *iteration* |
|---|---|---|---|---|---|---|---|---|
| (a1) | 10.8296 | 12.8007 | 10.3078 | 11.7105 | -0.30866 | -0.27677 | -0.046444 | 15 |
| (b1) | 12.6521 | 11.4687 | 12.4211 | 10.5745 | -0.02437 | 0.013397 | 0.047768 | 16 |
| (c1) | 12.0863 | 11.9282 | 10.6783 | 10.8417 | 0.056766 | 0.040253 | 0.08302 | 29 |
| (d1) | 10.2328 | 13.4534 | 10.8709 | 13.7534 | -0.69681 | -1.2756 | -0.14557 | 2 |
| (e1) | 11.8067 | 12.2777 | 10.3905 | 13.7904 | 0.044927 | 0.20894 | 0.18267 | 4 |
| (f1) | 12.2449 | 11.0597 | 12.8937 | 10.9927 | 1.0194 | 1.0461 | 0.20533 | 9 |
| (g1) | 9.9486 | 11.1448 | 11.1982 | 12.6798 | 0.02187 | 0.058481 | 0.089006 | 14 |
| (h1) | 20.1165 | 14.4592 | 10.164 | 4.4383 | 1.9137 | 0.89338 | 0.69462 | 4 |
| (i1) | 1.1724 | 11.1258 | 7.0007 | 9.2039 | -1.1153 | -0.35105 | -0.094687 | 10 |

**TABLE 11.** The fitness values of GLGSA and parameters of SPCNN for the images in Figure 8.

| *Image* | *CEP* | *EM* | *NC* | *fitness* | $\alpha_f$ | $\beta$ | $\alpha_\theta$ | $V_\theta$ | *iteration* |
|---|---|---|---|---|---|---|---|---|---|
| (a1) | 0.90309 | 0.43673 | 0.88245 | 0.74076 | -6.6296 | 4.934 | -5.7983 | -7.4506 | 2 |
| (b1) | 0.95508 | 0.57097 | 0.98439 | 0.83682 | 2.8086 | 3.1546 | 1.0183 | 1.5743 | 2 |
| (c1) | 0.90108 | 0.40589 | 0.98193 | 0.76297 | 4.2251 | -7.7609 | 1.0404 | 4.8254 | 5 |
| (d1) | 0.85646 | 0.52371 | 0.83906 | 0.73974 | 6.7235 | 7.6888 | 9.1406 | 2.9502 | 3 |
| (e1) | 0.87271 | 0.43786 | 0.94391 | 0.75149 | 8.8939 | 9.2712 | 9.5333 | 4.9346 | 3 |
| (f1) | 0.84844 | 0.65522 | 0.92772 | 0.81046 | -7.7915 | -9.4227 | -6.7286 | 7.4016 | 2 |
| (g1) | 0.87663 | 0.54553 | 0.82286 | 0.74834 | -7.3728 | 2.3366 | -7.2701 | -6.3509 | 12 |
| (h1) | 0.93149 | 0.44684 | 0.93114 | 0.77982 | 7.0068 | -0.54198 | 2.2055 | 4.3808 | 4 |
| (i1) | 0.90409 | 0.39441 | 0.96237 | 0.75362 | 6.5472 | 6.446 | 9.6397 | 1.8517 | 3 |

**TABLE 12.** The UM CM and SSIM values of the segmented images in Figure 9 using four methods, respectively.

| Method | Index | Image | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (a1) | (b1) | (c1) | (d1) | (e1) | (f1) | (g1) | (h1) | (i1) |
| *Otsu* | *UM* | 0.98698 | 0.98397 | 0.98687 | 0.98768 | 0.98478 | 0.98495 | 0.98984 | 0.99054 | 0.99039 |
| | *CM* | 0.38201 | 0.48824 | 0.3148 | 0.42567 | 0.40198 | 0.57903 | 0.32743 | 0.39438 | 0.5075 |
| | *SSIM* | 0.13839 | 0.7557 | 0.4002 | 0.33798 | 0.30794 | 0.31117 | 0.24435 | 0.54636 | 0.11638 |
| | *CE* | 1.50738 | 2.22791 | 1.70187 | 1.75133 | 1.6947 | 1.87515 | 1.56162 | 1.93128 | 1.61427 |
| *GA Renyi* | *UM* | 0.98697 | 0.99308 | 0.98686 | 0.98761 | 0.95496 | 0.98354 | 0.98599 | 0.99 | 0.98814 |
| | *CM* | 0.37967 | 0.54200 | 0.31492 | 0.42697 | 0.44695 | 0.55792 | 0.32887 | 0.4098 | 0.44671 |
| | *SSIM* | 0.21242 | 0.77079 | 0.40125 | 0.34485 | 0.40943 | 0.2903 | 0.3415 | 0.51825 | 0.056825 |
| | *CE* | 1.57906 | 2.30587 | 1.70303 | 1.75943 | 1.81134 | 1.83176 | 1.65636 | 1.91805 | 1.491675 |
| *PSO-PCNN* | *UM* | 0.99088 | 0.97245 | 0.98305 | 0.98058 | 0.97506 | 0.98197 | 0.99086 | 0.98539 | 0.98531 |
| | *CM* | 0.38140 | 0.19200 | 0.3103 | 0.43154 | 0.40611 | 0.61949 | 0.32833 | 0.2626 | 0.496 |
| | *SSIM* | 0.22033 | 0.39606 | 0.40244 | 0.35428 | 0.431 | 0.33121 | 0.33392 | 0.36953 | 0.11096 |
| | *CE* | 1.59261 | 1.56051 | 1.69579 | 1.7664 | 1.81217 | 1.93267 | 1.65311 | 1.61752 | 1.59227 |
| *Ours* | *UM* | 0.99558 | 0.99882 | 0.98995 | 0.98335 | 0.97664 | 0.98169 | 0.99416 | 0.98958 | 0.99528 |
| | *CM* | 0.40089 | 0.5835 | 0.30976 | 0.49251 | 0.4153 | 0.6342 | 0.40897 | 0.39423 | 0.49756 |
| | *SSIM* | 0.27032 | 0.77467 | 0.43597 | 0.42107 | 0.43824 | 0.36377 | 0.39516 | 0.58777 | 0.12972 |
| | *CE* | 1.66679 | 2.35699 | 1.73568 | 1.89693 | 1.83018 | 1.97966 | 1.79829 | 1.97158 | 1.62256 |

where $C$ denotes the normalization factor, $R_j$ and $A_j$ represent the $j$th region and the corresponding number of pixels in $j$th region respectively. $n$ is the number of regions, $i$ is the number of gray level which is set to 2 in the binary image. The larger the *UM* value indicates that the effect of image segmentation is better.

The CM shows the contrast between the mean of the background and the target pixel, which is described as:

$$CM = \frac{|f_o - f_b|}{f_o + f_b} \tag{38}$$

here $f_o$ and $f_b$ are the average value of the object and background, respectively. The larger the CM value represents the higher the accuracy of image segmentation.

The SSIM is used to calculate the structural similarity of the original image and the segmented image, which is defined as:

$$SSIM(I, I') = \frac{(2\mu_I\mu_{I'} + c_1)(2\sigma_{II'} + c_2)}{(\mu_I^2 + \mu_{I'}^2 + c_1)(\sigma_I^2 + \sigma_{I'}^2 + c_2)} \tag{39}$$

where $\mu_I$ and $\mu_{I'}$ are the the mean value of $I$ and $I'$ respectively. $\sigma_I^2$ and $\sigma_{I'}^2$ are the variance of $I$ and $I'$, $\sigma_{II'}$ denotes the covariance of $I$ and $I'$, the two variables $c_1$=$(k_1L)^2$ and $c_2 = (k_2L)^2$ stabilize the division with weak denominator, $L$ represents the dynamic range of pixel values, $k_1 = 0.01$ and $k_2 = 0.03$ by default.

Quantitative evaluation of the image segmentation is shown in Table 12, in order to analyze the results more visually and intuitively, the data of Table 12 is represented in Figure 9–14. In the view of UM and CM, although we could find that the performance of our proposed mothed is not the best for all the image from Table 12 and Figure 9 and 10,

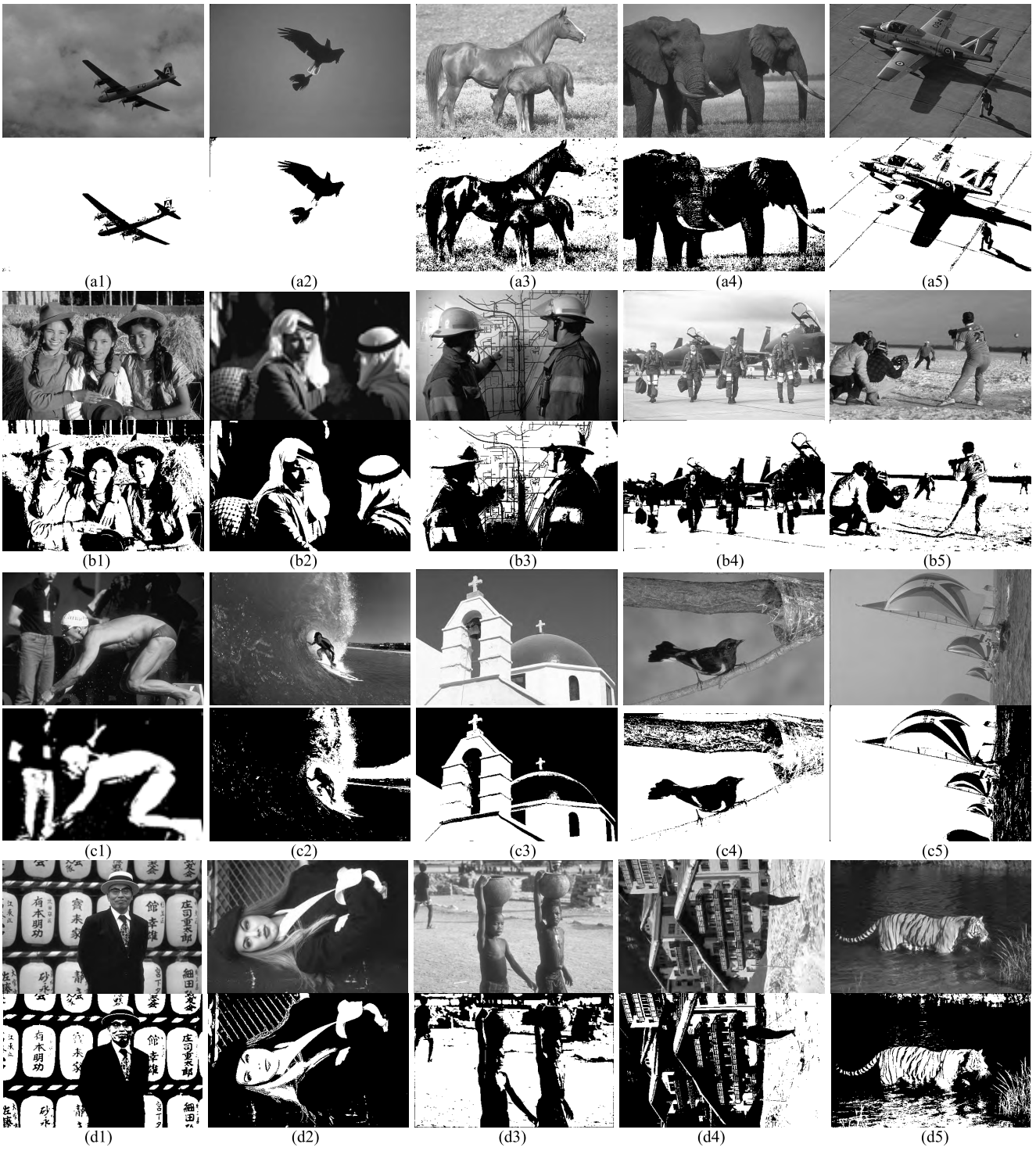


**FIGURE 15.** Some original and segmented images.

our proposed method and the classic Otsu are very steady, with no very poor results. From the perspective of SSIM, our proposed method is very competitive with the other three methods in Table 12 and Figure 11, which indicates that the EM is very useful as the part of fitness. In the standpoint of CE, the method we propose is the best and has great advantage from Figure 12 and 13. From viewpoint of the average values of UM, CM and SSIM corresponding to all images in Figure 14, our method is steady and has a good performance. More images and segmentation results are shown in Figure 15, whose fitness and SPCNN parameters are put in Table 13.

**TABLE 13.** The fitness values of GLGSA and parameters of SPCNN for the images in Figure 15.

| *Image* | *CEP* | *EM* | *NC* | *fitness* | $\alpha_f$ | $\beta$ | $\alpha_\theta$ | $V_\theta$ | *iteration* |
|---|---|---|---|---|---|---|---|---|---|
| (a1) | 0.98683 | 0.54873 | 0.99806 | 0.84454 | 1.6634 | -4.0886 | 0.93827 | 5.1452 | 2 |
| (a2) | 0.96115 | 0.62587 | 0.87247 | 0.81983 | -2.7755 | -5.6352 | -2.3831 | 5.8445 | 2 |
| (a3) | 0.94638 | 0.54014 | 0.90389 | 0.7968 | 5.6582 | 9.772 | 0.9547 | 3.5474 | 2 |
| (a4) | 0.94768 | 0.42476 | 0.91244 | 0.76162 | -0.56496 | -3.9863 | 6.9997 | 5.2521 | 7 |
| (a5) | 0.96496 | 0.58352 | 0.99184 | 0.84677 | -2.8635 | -5.6553 | -2.5981 | 2.6407 | 2 |
| (b1) | 0.83885 | 0.56477 | 0.94275 | 0.78212 | -6.6944 | -0.81947 | -6.2196 | 4.9742 | 3 |
| (b2) | 0.86391 | 0.45699 | 0.98439 | 0.76843 | -0.95364 | -0.036191 | -0.10275 | 2.6985 | 2 |
| (b3) | 0.81665 | 0.62827 | 0.95045 | 0.79846 | -1.2517 | -0.90445 | -0.46513 | 8.8423 | 2 |
| (b4) | 0.96448 | 0.61767 | 0.97519 | 0.85245 | -3.4506 | 8.2844 | -3.3503 | -6.5379 | 5 |
| (b5) | 0.93046 | 0.47415 | 0.94883 | 0.78448 | -6.6296 | 4.934 | -5.7983 | -7.4506 | 2 |
| (c1) | 0.86887 | 0.39717 | 0.98918 | 0.75174 | -4.3981 | -7.4494 | -3.8882 | 2.1855 | 3 |
| (c2) | 0.91416 | 0.43086 | 0.95492 | 0.76665 | -2.1555 | -4.9783 | -2.0043 | 4.6344 | 5 |
| (c3) | 0.97579 | 0.56924 | 0.99417 | 0.8464 | 6.2286 | 5.7818 | 3.1605 | 5.7592 | 4 |
| (c4) | 0.9749 | 0.65616 | 0.96289 | 0.86465 | -7.9937 | 7.0781 | -6.6362 | 0.90667 | 2 |
| (c5) | 0.98208 | 0.60656 | 0.95415 | 0.8476 | -9.808 | -6.6667 | -9.4302 | 2.7656 | 3 |
| (d1) | 0.90341 | 0.65378 | 0.98821 | 0.84847 | -2.1555 | -4.9783 | -2.0043 | 4.6344 | 6 |
| (d2) | 0.91815 | 0.5145 | 0.97468 | 0.80244 | 4.5031 | 1.2144 | 2.8256 | 0.89456 | 7 |
| (d3) | 0.94834 | 0.53807 | 0.9796 | 0.822 | 0.01458 | 3.8173 | 8.2392 | 7.4298 | 5 |
| (d4) | 0.85552 | 0.67607 | 0.92092 | 0.8175 | -0.08509 | -7.1892 | -0.06099 | 0.92536 | 2 |
| (d5) | 0.92731 | 0.61834 | 0.90965 | 0.81843 | 2.8086 | 3.1546 | 1.10183 | 1.5743 | 2 |

## VI. CONCLUSION

In this work, a novel image segmentation method was proposed based on SPCNN, which was optimized by GLGSA that combined GSA with gbest agent memory ability. GLGSA proved its better performance on 23 standard benchmark functions in terms of convergence rate and avoidance of local minim compared with GSA and GGSA. In order to make the image segmentation more accurate, the fitness function consisted of CEP, EM and NC. In the view of the subjective visual analysis and quantitative analysis, our proposed method could get better segmentation results in comparision with the state-of-the-art algorithm, such as Otsu, GA Renyi and PSO-PCNN, using the gray nature images from the Berkeley Segmentation Dataset.

Our future work would improve the convergence rate and avoid local minim, which would use the hybrid of GSA and chaos. At the same time, we would apply the PCNN to do some research on color image segmentation.

## REFERENCES

[1] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[2] H. Yu, F. He, and Y. Pan, "A novel region-based active contour model via local patch similarity measure for image segmentation," *Multimedia Tools Appl.*, vol. 77, no. 18, pp. 24097–24119, Sep. 2018.

[3] A. Fabijańska, "Variance filter for edge detection and edge-based image segmentation," in *Proc. MEMSTECH*, Polyana Ukraine, May 2011, pp. 151–154.

[4] M. Z. Abderrezak, M. B. Chibane, and K. Mansour, "A new segmentation method of cerebral MRI images based on the fuzzy C-means algorithm," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 25, no. 4, pp. 3215–3226, Jul. 2017.

[5] L.-H. Li, B. Qian, J. Lian, W.-N. Zheng, and Y.-F. Zhou, "Study on semantic image segmentation based on convolutional neural network," *J. Intell. Fuzzy Syst.*, vol. 33, no. 6, pp. 3397–3404, Nov. 2017.

[6] A. D. Gritzman, V. Aharonson, D. M. Rubin, and A. Pantanowitz, "Automatic computation of histogram threshold for lip segmentation using feedback of shape information," *Signal, Image Video Process.*, vol. 10, no. 5, pp. 869–876, Jul. 2016.

[7] L. He, X. Yu, Z. Huang, and A. M. A. Talab, "The two-dimensional double-entropy threshold based on the parallel genetic simulated annealing algorithms," *Optik*, vol. 127, no. 1, pp. 96–101, Jan. 2016.

[8] Y. Chen and C. Han, "A modified region growing algorithm for multi-colored image object segmentation," *Chin. Opt. Lett.*, vol. 5, no. 1, pp. 25–27, Jan. 2007.

[9] Y. Pan, T. Zhou, and Y. Xi, "Bacterial foraging based edge detection for cell image segmentation," in *Proc. EMBC*, Milan, Italy, Aug. 2015, pp. 3873–3876.

[10] J. Zheng, D. Zhang, K. Huang, and Y. Sun, "Adaptive image segmentation method based on the fuzzy c-means with spatial information," *IET Image Process.*, vol. 12, no. 5, pp. 785–792, May 2018.

[11] M. Buscema, "Back propagation neural networks," *Substance Use Misuse*, vol. 33, no. 2, pp. 233–270, Jan. 1998.

[12] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comput.*, vol. 13, no. 2, pp. 246–257, Jun. 1991.

[13] R. Eckhorn, H. J. Reitboeck, M. Arndt, and P. Dicke, "Feature linking via synchronization among distributed assemblies: Simulations of results from cat visual cortex," *Neural Comput.*, vol. 2, no. 3, pp. 293–307, Sep. 1990.

[14] J. L. Johnson and M. L. Padgett, "PCNN models and applications," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 480–498, May 1999.

[15] Y. Chen, S. K. Park, Y. Ma, and R. Ala, "A new automatic parameter setting method of a simplified PCNN for image segmentation," *IEEE Trans. Neural Netw.*, vol. 22, no. 6, pp. 880–892, Jun. 2011.

[16] Z. Wang, S. Wang, Y. Zhu, and Y. Ma, "Review of image fusion based on pulse-coupled neural network," *Arch. Comput. Methods Eng.*, vol. 23, no. 4, pp. 659–671, Dec. 2016.

[17] X. Gu, "Feature extraction using unit-linking pulse coupled neural network and its applications," *Neural Process. Lett.*, vol. 27, no. 1, pp. 25–41, Feb. 2008.

[18] D. Zhou and Y. Shao, "Region growing for image segmentation using an extended PCNN model," *IET Image Process.*, vol. 12, no. 5, pp. 729–737, May 2018.

[19] F. He, Y. Guo, and C. Gao, "An improved pulse coupled neural network with spectral residual for infrared pedestrian segmentation," *Infr. Phys. Technol.*, vol. 87, pp. 22–30, Dec. 2017.

[20] J. Lian, B. Shi, M. Li, Z. Nan, and Y. Ma, "An automatic segmentation method of a parameter-adaptive PCNN for medical images," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 9, pp. 1511–1519, Sep. 2017.

[21] J. Zhang, F. Kong, Z. Zhai, J. Wu, and S. Han, "Robust image segmentation method for cotton leaf under natural conditions based on immune algorithm and PCNN algorithm," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 5, pp. 1854011-1–1854011-23, 2018.

[22] X. Xu, T. Liang, G. Wang, M. Wang, and X. Wang, "Self-adaptive PCNN based on the ACO algorithm and its application on medical image segmentation," *Intell. Automat. Soft Comput.*, vol. 23, no. 2, pp. 303–310, Jul. 2016.

[23] I. S. Hage and R. F. Hamade, "Segmentation of histology slides of cortical bone using pulse coupled neural networks optimized by particle-swarm optimization," *Computerized Med. Imag. Graph.*, vol. 37, nos. 7–8, pp. 466–474, Oct./Dec. 2013.
[24] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.
[25] C. Gupta and S. Jain, "Multilevel fuzzy partition segmentation of satellite images using GSA," in *Proc. ICSPCT*, Ajmer, India, Jul. 2014, pp. 173–178.
[26] M. B. Dowlatshahi and H. Nezamabadi-Pour, "GGSA: A Grouping Gravitational Search Algorithm for data clustering," *Eng. Appl. Artif. Intell.*, vol. 36, pp. 114–121, Nov. 2014.
[27] R.-E. Precup, R.-C. David, E.-M. Petriu, S. Preitl, and M.-B. Radac, "Fuzzy logic-based adaptive gravitational search algorithm for optimal tuning of fuzzy-controlled servo systems," *IET Control Theory Appl.*, vol. 7, no. 1, pp. 99–107, Jan. 2013.
[28] S. Mirjalili, S. Z. M. Hashim, and H. M. Sardroudi, "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm," *Appl. Math. Comput.*, vol. 218, no. 22, pp. 11125–11137, Jul. 2012.
[29] G. Sun and A. Zhang, "A hybrid genetic algorithm and gravitational search algorithm for image segmentation using multilevel thresholding," in *Proc. ibPRIA*, Madeira, Portugal, 2013, pp. 707–714.
[30] S. Gao, C. Vairappan, Y. Wang, Q. Cao, and Z. Tang, "Gravitational search algorithm combined with chaos for unconstrained numerical optimization," *Appl. Math. Comput.*, vol. 231, pp. 48–62, Mar. 2014.
[31] J. L. Johnson and D. Ritter, "Observation of periodic waves in a pulse-coupled neural network," *Opt. Lett.*, vol. 18, no. 15, pp. 1253–1255, Aug. 1993.
[32] C. Gao, D. Zhou, and Y. Guo, "An iterative thresholding segmentation model using a modified pulse coupled neural network," *Neural Process. Lett.*, vol. 39, no. 1, pp. 81–95, Feb. 2014.
[33] C. Kavitha, C. Chellamuthu, and R. Rajesh, "Multimodal medical image fusion using discrete ripplet transform and intersecting cortical model," *Proc. Eng.*, vol. 38, pp. 1409–1414, Jun. 2012.
[34] H. Li, D. Xu, and R. Zong, "Face recognition based on unit-linking PCNN time signature," in *Proc. ICACC*, Singapore, Jan. 2009, pp. 360–364.
[35] H. Yamada, Y. Ogawa, K. Ishimura, and M. Wada, "Face detection using pulse-coupled neural network," in *Proc. SICE*, Fukui, Japan, Aug. 2003, pp. 2784–2788.
[36] K. Zhan, H. Zhang, and Y. Ma, "New spiking cortical model for invariant texture retrieval and image processing," *IEEE Trans. Neural Netw.*, vol. 20, no. 12, pp. 1980–1986, Dec. 2009.
[37] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. MHS*, Nagoya, Japan, Oct. 1995, pp. 39–43.
[38] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.
[39] S. Mirjalili and A. Lewis, "Adaptive gbest-guided gravitational search algorithm," *Neural Comput. Appl.*, vol. 25, nos. 7–8, pp. 1569–1584, Dec. 2014.
[40] S. Abdel-Khalek, A. B. Ishak, O. A. Omer, and A.-S. F. Obada, "A two-dimensional image segmentation method based on genetic algorithm and entropy," *Optik*, vol. 131, pp. 414–422, Feb. 2017.
[41] X. Xinzheng, D. Shifei, S. Zhongzhi, Z. Hong, and Z. Zuopeng, "Particle swarm optimization for automatic parameters determination of pulse coupled neural network," *J. Comput.*, vol. 6, no. 8, pp. 1546–1553, Aug. 2011.
[42] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. ICCV*, Vancouver, BC, Canada, Jul. 2001, pp. 416–423.
[43] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Comput. Vis., Graph., Image Process.*, vol. 41, no. 2, pp. 233–260, Feb. 1988.
[44] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

**KEMING JIAO** received the B.S. degree in electrical engineering and automation from Chongqing Jiaotong University, Chongqing, China, in 2015. He is currently pursuing the M.S. degree in circuits and systems with the School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou, China. His research interests include intelligent control and image processing.

**ZHONGLIANG PAN** received the M.S. degree from Tsinghua University, Beijing, China, in 1991, and the Ph.D. degree in circuits and systems from the University of Electronic Science and Technology of China, Chengdu, China, in 1997. He was a Postdoctoral Research Associate with Sun Yat-sen University, Guangzhou, China, from 1998 to 1999. Since 2004, he has been a Professor with the School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou. His current research interests include image processing and electronic design automation and testing.