

## Research Article

# Genetic Pattern Search and Its Application to Brain Image Classification

Yudong Zhang,<sup>1</sup> Shuihua Wang,<sup>1,2</sup> Genlin Ji,<sup>1</sup> and Zhengchao Dong<sup>3</sup>

<sup>1</sup> School of Computer Science and Technology, Nanjing Normal University, Nanjing, Jiangsu 210023, China

<sup>2</sup> School of Electronic Science and Engineering, Nanjing University, Nanjing, Jiangsu 210046, China

<sup>3</sup> Brain Imaging Laboratory & MRI Unit, Columbia University and New York State Psychiatric Institute, New York, NY 10032, USA

Correspondence should be addressed to Yudong Zhang; [zhangyudongnuaa@gmail.com](mailto:zhangyudongnuaa@gmail.com)

Received 1 July 2013; Revised 20 August 2013; Accepted 7 September 2013

Academic Editor: Vishal Bhatnagar

Copyright © 2013 Yudong Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel global optimization method, based on the combination of genetic algorithm (GA) and generalized pattern search (PS) algorithm, is proposed to find global minimal points more effectively and rapidly. The idea lies in the facts that GA tends to be quite good at finding generally good global solutions, but quite inefficient in finding the last few mutations for the absolute optimum, and that PS is quite efficient in finding absolute optimum in a limited region. The novel algorithm, named as genetic pattern search (GPS), employs the GA as the search method at every step of PS. Experiments on five different classical benchmark functions (consisting of Hump, Powell, Rosenbrock, Schaffer, and Woods) demonstrate that the proposed GPS is superior to improved GA and improved PS with respect to success rate. We applied the GPS to the classification of normal and abnormal structural brain MRI images. The results indicate that GPS exceeds BP, MBP, IGA, and IPS in terms of classification accuracy. This suggests that GPS is an effective and viable global optimization method and can be applied to brain MRI classification.

## 1. Introduction

The evolutionary computation community has shown for many years significant interest in optimization problems, in particular in the global optimization of numerical, real-valued “black-box” problems, for which exact and analytical methods are not productive. Genetic algorithm (GA) [1], generalized pattern search (PS) [2], particle swarm optimization (PSO) [3], firefly algorithm [4], and differential evolution (DE) [5] are among the most recent developments. These techniques have shown great promise in several real-world applications.

In most cases, an optimization problem is divided into two phases: the coarse search and the fine search [6]. GA is well suited for a swift and global exploration of a large search space to optimize an objective function and to target the region near the optimum point quickly [7]. However, GA may run with difficulty in the immediate vicinity of the optimum point [8]. Conversely, PS is a nonrandom method to search for minima of a function that is not differentiable,

stochastic, or even continuous without requiring the gradient information [9]. It performs especially well in local search, but it is sensitive for the randomly or manually input initial values, and requires a high degree of expertise by the user [10].

The complementary advantages of GA and PS motivated our strategy in this paper that combines both GA and PS to produce a new algorithm referred to as genetic pattern search (GPS). We evaluated the proposed method by five different classical benchmark functions (consisting of Hump, Powell, Rosenbrock, Schaffer, and Woods) and applied it to the classification of normal and abnormal brain MRI images.

The structure of this paper was organized as follows. Section 2 gave a detailed introduction to GA and PS, respectively. Section 3 outlined the structure and flow of the proposed GPS algorithm. Experiments in Section 4 compared GPS with improved GA and improved PS on five test functions. Section 5 applied the GPS to structural brain image classification. Finally Section 6 concluded this paper.

## 2. Background

*2.1. Principles of GA.* GAs are powerful stochastic search techniques based on the processes of natural selection [11]. These techniques perform heuristic search that mimics the process of natural evolution, such as inheritance, mutation, selection, and crossover. This heuristic is routinely used to generate useful solutions to optimization and search problems [12]. The principles of the GAs can be described as follows.

The trial solutions of GAs are encoded in the form of strings. Each string is associated with an objective function that represents the degree of the fitness of the string. A collection of such strings is called a population. A random population with a few strings with higher fitness is initially created to represent different points in the search space. Each of these strings is assigned a number of copies that go into the mating pool, based on the principle of survival of the fittest [13]. Crossover and mutation operators are applied on these strings. The processes of selection, crossover, and mutation continue until either a fixed number of generations or a termination condition is reached [14]. The above procedures of GA can be realized by the following pseudocode.

- (1) Choose the initial population of individuals.
- (2) Evaluate the fitness of each individual in that population.
- (3) Repeat on this generation until termination criteria are met.
  - (A) Select the best-fit individuals for reproduction.
  - (B) Breed new individuals through crossover and mutation operations to give birth to offspring.
  - (C) Evaluate the individual fitness of new individuals.
  - (D) Replace the least-fit population with new individuals.

GAs differ from classical optimization techniques such as the gradient-based algorithm in the following three ways: (1) GAs work on a population of points instead of a single point; (2) GAs use only the values of the objective function not their derivatives or other auxiliary knowledge; (3) GAs use probabilistic transition functions and not deterministic ones [15].

*2.2. Introduction of PS.* PS is a method that updates current iterate by sampling the objective function to find a decrease at a finite number of points along a suitable set of search directions [16]. Suppose that;  $f$  denotes the objective function, starting from an initial guess  $x_0$  and initial step length  $\Delta_0$ , the PS generates a sequence of iterates such that  $f(x_{k+1}) \leq f(x_k)$ . Each iteration consists of a “search” step and a “poll” step. The search step generates a finite number of trial points on a mesh  $M_k$ , which is centered at  $x_k$  and defined by a finite set of patterns  $\Gamma$ , which positively span the solution space. The mesh is given by

$$M_k(i) = \{x_k + \Delta_k d_i, d_i \in \Gamma\}. \quad (1)$$

The poll step polls the points in the current mesh to find a better one. The polling can be either complete, meaning that all points are polled, or incomplete, namely, the algorithm stops polling as soon as it finds a point whose objective function value is less than the current point. The complete poll performs better but consumes more time. The incomplete poll finds the local optima [17].

If the poll step finds an improved point, then  $x_{k+1}$  equals to the new point, and update the step length by multiplying  $\Delta_k$  by expansion factor  $\alpha_e (>1)$  such that  $\Delta_{k+1} > \Delta_k$  because current pattern is a suitable set of poll directions. Otherwise, the poll step cannot find an improved point; then  $x_{k+1} = x_k$ , and the step length is reduced by multiplying  $\Delta_k$  by contraction factor  $\alpha_c (<1)$  such that  $\Delta_{k+1} < \Delta_k$ . The aforementioned description can be summarized as follows.

- (1) Initialization: generate patterns  $\Gamma$  and initialize the step length  $\Delta_0$ .
- (2) Repeat on this generation until termination criteria are met:
  - (A) generate new mesh points  $\{M_k\}$ ;
  - (B) poll the mesh points. If successful, expand the mesh; otherwise contract the mesh.

## 3. Genetic Pattern Search

The proposed genetic pattern search (GPS) combines both GA and PS algorithms. In 2011, Zhang et al. have proposed a combination based on GA and PS [18]; however, in their method the combination algorithm was done by a first stage using a GA followed by a second stage that uses PS taking as an input the output of the GA. The method has a serious drawback: the PS could start when the GA has not reached the neighborhood of the global optimum yet. In that way, we would not take full advantage of the GA. This drawback will become more critical when the problem to be solved becomes more complex.

Therefore, in this paper, we take the GA as the search method at every step of PS. In this way, the coarse and fine search would be made in each epoch, and it could speed the convergence up as shown in Figure 1. The pseudocodes of GPS are described as follows.

Step 1 (initialization). Generate pattern  $\Gamma = \{d_1, d_2, d_3, \dots, d_{2N}\}$ , initial point  $x_0$ , initialize step length  $\Delta_0$ , and let  $k = 0$ .

Step 2. Generate new mesh points  $\{M_k(i)\} = x_k + \Delta_k \Gamma$ .

Step 3. GA Search with initial population designed as  $\{M_k(i)\}$ . The mesh points are updated as the final GA results  $\{M_k^U(i)\}$ .

Step 4 (complete poll). If  $i^*$  exists so that  $f[M_k^U(i^*)] < f(x_k)$ , then it is a successful poll. Let  $x_{k+1} = M_k^U(i^*)$ , and  $\Delta_{k+1} = \alpha_e \Delta_k$ . Otherwise, the poll cannot find an improved point; let  $x_{k+1} = x_k$ , and  $\Delta_{k+1} = \alpha_c \Delta_k$ .

Step 5 ( $k = k + 1$ ). Check whether the termination conditions are satisfied. If yes, output  $x_k$ , otherwise jump to Step 2.

TABLE 1: Information of five benchmark functions.

Test function	Function formulation	Optimal point
Hump	$4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4 + 2$	$F(-0.0898, 0.7126) =$ $F(0.0898, -0.7126) =$ 0.9684
Powell	$(x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - x_3)^4 + 10(x_1 - x_4)^4$	$F(0, 0, 0, 0) = 0$
Rosenbrock	$100(x_1^2 - x_2)^2 + (1 - x_1)^2$	$F(1, 1) = 0$
Schaffer	$\frac{[\sin(x_1^2 + x_2^2)]^{1/2}]^2 - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} + 1.5$	$F(0, 0) = 1$
Woods	$100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$	$F(1, 1, 1, 1) = 0$

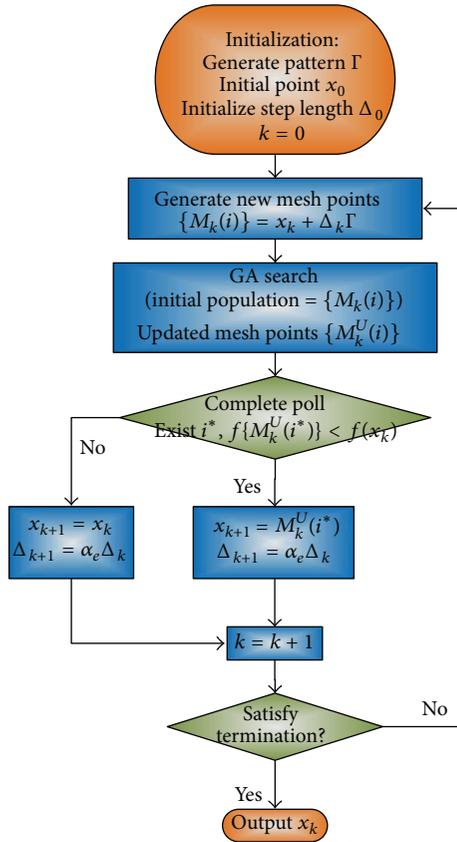


FIGURE 1: Flowchart of GPS algorithm.

## 4. Experiments

The test and evaluation were performed with an IBM P4 with 2 GHz processor and 1 GB memory. The program is in-house developed by MATLAB 2010b. The readers can reproduce the work at any laptop installing MATLAB.

**4.1. Test Functions.** We used five classical benchmark functions to evaluate the performance of the GPS algorithm. Among the five benchmark functions, Hump, Rosenbrock, and Schaffer functions are two-dimensional and Powell and Woods functions are four-dimensional. Formulations, global

optimal points, and fitness values of those functions are listed in Table 1.

It should be noted that the constant terms are added (2 in Hump function and 1.5 in Schaffer function) to make sure that the range of fitness value is always above zero so that the  $y$ -axis can be plotted in logarithmic scale. Furthermore, the initial points of each test function were selected far from the optimal point deliberately to test the robustness of the algorithms. Figure 2 shows the surface plot and contour lines of Hump, Rosenbrock, and Schaffer functions, respectively. The Powell and Woods functions are not displayed due to their high-dimensional property.

**4.2. Parameter Setting.** An improved GA (IGA) [19] and an improved PS (IPS) [20] were chosen for comparison. The parameters of IGA and IPS were obtained by numerous experiments, and the parameters corresponding to the best results are selected and listed in the first and second rows in Table 2. Besides, IGA and IPS are both evolutionary computations, which features in robustness of results with variation of parameters. Coelho et al. [21] and Ghanbari and Mahdavi-Amiri [22] have already proved that the results of evolutionary algorithms are sensitive to neither initial values nor parameter values. Therefore, the results of both IGA and IPS will exhibit little variation even if their corresponding parameters are changed far from the values determined in this experiment. The parameters of GPS were set by combining the parameters of IGA and IPS.

The initial population of IGA was set randomly distributed in the whole search space. For the deterministic algorithm IPS, we determine the initial point changed randomly in each run. In this experiment, we set the termination criteria as in the last row of Table 2 by trial-and-error method.

**4.3. Success Rate Comparison.** The success rate is the rate at which the algorithm converges to the nearly absolute global optimal point. Suppose that  $x^*$  denotes the global optimal point, and  $x$  denotes the point found in a run. A more accurate definition of success rate is shown as follows:

$$\text{if } \frac{|x - x^*|}{\max(1, |x^*|)} < 0.001, \quad \text{then } x \text{ is success.} \quad (2)$$

The above formula means that the found solution  $x$  is treated identical to the global optima  $x^*$  if the norm of their

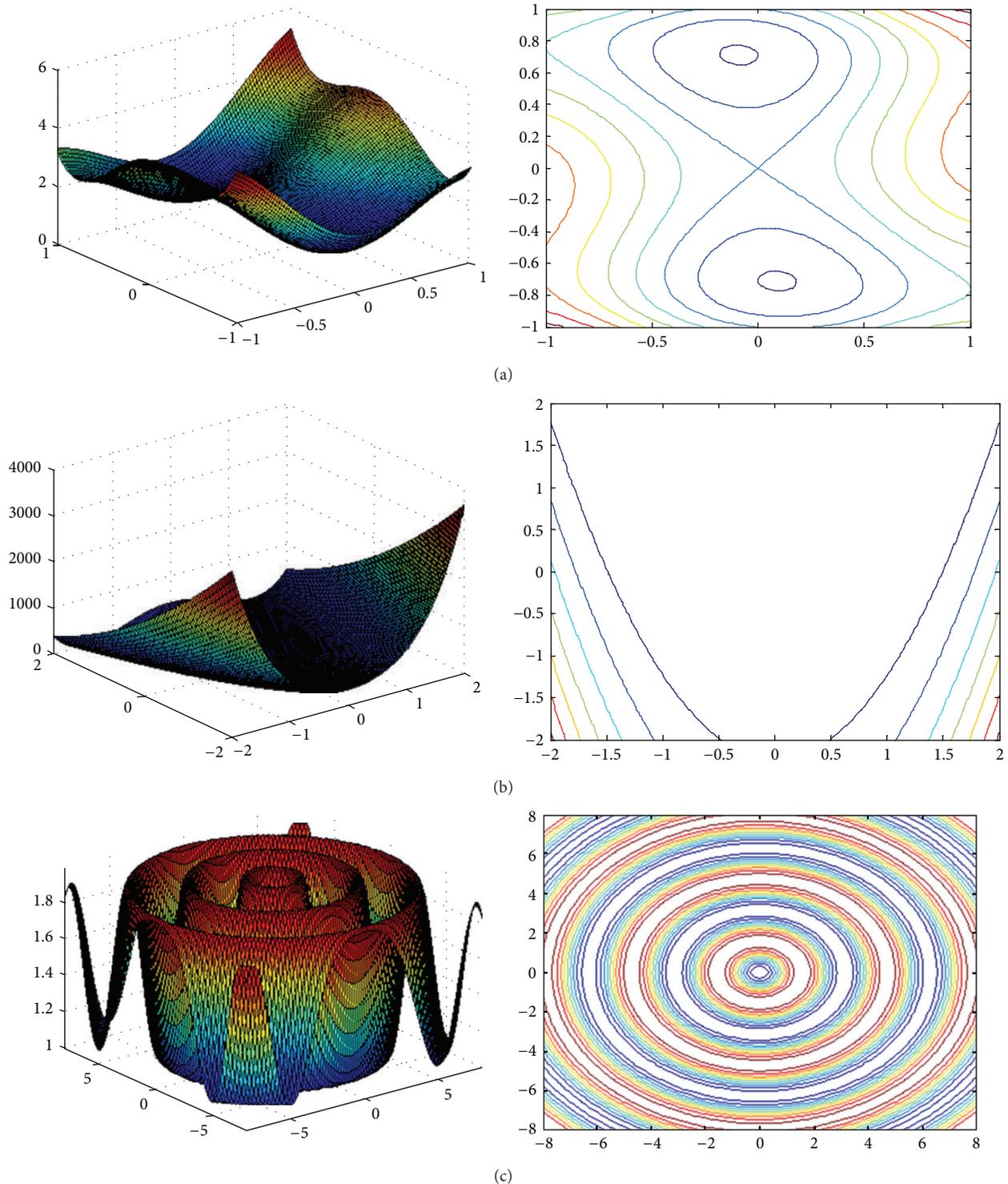


FIGURE 2: Surface plot and contour lines of 2D test functions: (a) Hump; (b) Rosenbrock; (c) Schaffer.

differences is less than 0.1% of the maximum between 1 and the norm of  $x^*$ . The reason why we do not use the rigid equation  $x = x^*$  is that finding the accurate global optimal is impossible. The reason lies in the following 3 points: (I) there are always round-up errors during the computation; (II) the word length of the computer is limited; (III) 0.1% relative error is enough in most of industrial applications.

After 100 runs of each algorithm, the success rate was calculated and listed in Table 3. The data here is not the same from [18] because the parameters are different: (I) here we set the maximum fitness evaluation as infinity while in [18] it is only  $2 \times 10^4$ ; (II) here the GPS is done by adding GA at each step of PS, while the algorithm is done by first GA followed by PS in [18]; (III) here the initialization of IPS is randomly

TABLE 2: Parameters setting.

IGA	Population = 20; elite number = 2; crossover rate = 0.8, mutation rate = 0.2, migration interval = 20; migration direction = forward; fitness scaling = rank; selection function = stochastic
IPS	Population = 2N; poll = 2N positive basis; complete poll = yes; initial mesh size = 1; expansion factor = 2.0; contraction factor = 0.5.
GPS	GA: population = 20; elite number = 2; crossover rate = 0.8, mutation rate = 0.2, fitness scaling = rank; selection function = stochastic. PS: population = 2N; Poll = 2N positive basis; complete poll = Yes; initial mesh size = 1; expansion factor = 2.0; contraction factor = 0.5.
Termination	Maximum fitness evaluate = $\infty$ ; independent variables tolerance = $10^{-4}$ ; fitness tolerance = $10^{-4}$ ; stall iteration = 20; mesh precision = $10^{-4}$ .

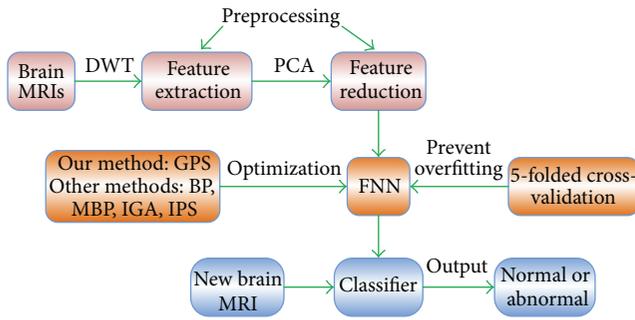


FIGURE 3: Flowchart of our brain classification system.

TABLE 3: Success rate comparison.

Test function	IGA	IPS	GPS
Hump	82	95	<b>98</b>
Powell	41	94	<b>99</b>
Rosenbrock	62	88	<b>95</b>
Schaffer	75	71	<b>96</b>
Woods	33	29	<b>85</b>

created at each run, while the initial values of IPS are fixed in [18].

For all functions, the GPS performs best with high enough success rates. For Hump, Powell, and Rosenbrock functions, the IPS is better than IGA. For Schaffer and Woods functions, the IGA is better than IPS. In total, the GPS is the most robust algorithm among the three.

### 5. Application

As an application, we applied the GPS algorithm to the weights optimization of forward neural network (FNN), which is used as a classifier of structural MRI images between normal and abnormal brains.

*5.1. Method.* The strategy is shown in Figure 3. First, we extract features via discrete wavelet transform (DWT), which has already proved to be an effective strategy for clinical diagnosis [23–25]. Second, the wavelet domain features are reduced via principle component analysis (PCA). Third, we use stratified  $K$ -fold cross-validation to prevent overfitting

of the following classifier. Fourth, the reduced features are sent to the FNN. Fifth, the GPS and other training algorithms are employed to train the FNN. Finally, the classification accuracies of FNNs trained by different algorithms are compared. The following paragraphs will discuss the procedures in detail.

The DWT is a powerful implementation of the wavelet transform using the dyadic scales and positions [26]. In this study, since the brain images are 2D data, the DWT is applied to horizontal and vertical dimensions separately. As a result, there are 4 subband images at each scale. The subband  $cA_{j+1}$  is used for next 2D DWT. As the level of decomposition increased, a more compact but coarser approximation component was obtained. Thus, wavelets provide a simple hierarchical framework for interpreting the image information.

PCA is an efficient tool to reduce the dimension of a data set consisting of a large number of interrelated variables while retaining most of the variations [27]. The PCA describes the space of the original data projecting onto the space in a base of eigenvectors. The corresponding eigenvalues account for the energy of the process in the eigenvector directions. It is assumed that most of the information in the observation vectors is contained in the subspace spanned by the first several PCs.

Cross-validation methods consist of three types: random subsampling,  $K$ -fold cross-validation, and leave-one-out validation. The  $K$ -fold cross-validation is applied due to its properties as being simple to learn, easy to realize, and using all data for training and validation. The mechanism is to create a  $K$ -fold partition of the whole dataset, repeat  $K$  times to use  $K - 1$  folds for training and a left fold for validation, and finally average the error rates of  $K$  experiments. The  $K$  folds can be randomly partitioned. However, some folds may have quite different distributions from other folds. Therefore, stratified  $K$ -fold cross-validation was employed, with which every fold has nearly the same class distributions [28]. In this study, we empirically determined  $K$  as 5 through the trial-and-error method.

FNNs are widely used in pattern classification since they do not need any information about the probability distribution and the a priori probabilities of different classes [29]. A single-hidden-layer backpropagation (BP) neural network is adopted with sigmoid neurons in the hidden layer and linear neuron in the output layer.

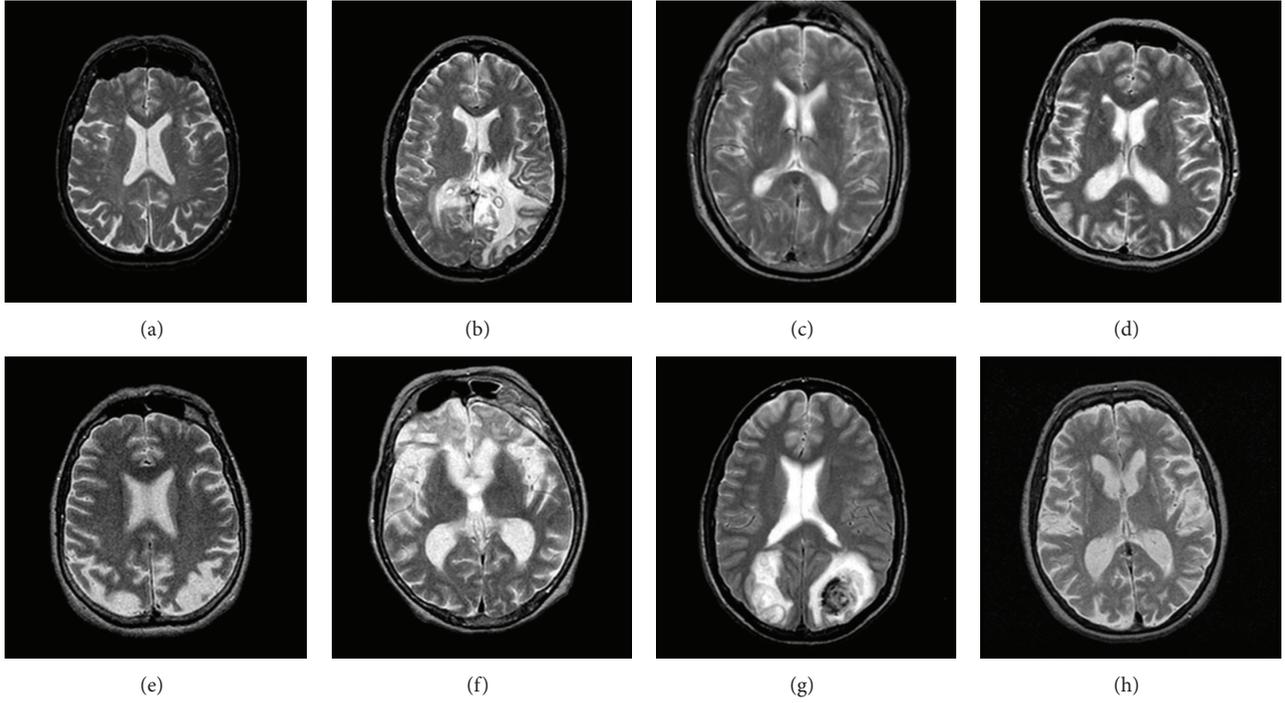


FIGURE 4: Sample of normal and abnormal brain images: (a) normal brain; (b) glioma; (c) meningioma; (d) Alzheimer's disease; (e) Alzheimer's disease with visual agnosia; (f) Pick's disease; (g) sarcoma; (h) Huntington's disease.

The training vectors were presented to the FNN, which is trained in batch mode. The network configuration is supposed as  $N_I \times N_H \times N_O$ , that is, a two-layer network with  $N_I$  input neurons,  $N_H$  neurons in the hidden layer, and  $N_O$  output indicating the brain is normal or abnormal. Assume that  $\omega_1$  and  $\omega_2$  represent the connection weight matrix between the input layer and hidden layer, between the hidden layer and the output layer, respectively, the outputs of all neurons in the hidden layer are calculated by

$$y_j = f_H \left( \sum_{i=1}^{N_I} \omega_1(i, j) x_i \right), \quad j = 1, 2, \dots, N_H. \quad (3)$$

Here,  $x_i$  denotes the  $i$ th input value,  $y_j$  denotes the  $j$ th output of the hidden layer, and  $f_H$  refers to the activation function of hidden layer, usually a sigmoid function. The outputs of all neurons in the output layer are given as follows:

$$O_k = f_O \left( \sum_{j=1}^{N_H} \omega_2(j, k) y_j \right), \quad k = 1, 2, \dots, N_O. \quad (4)$$

Here,  $f_O$  denotes the activation function of output layer, usually a line function. All weights are assigned to random values initially and are modified by the delta rule according to the learning samples. The error is expressed as the mean squared error (MSE) of the difference between output and target values:

$$E_l = \text{mse} \left( \sum_{k=1}^{N_O} (O_k - T_k) \right), \quad l = 1, 2, \dots, N_S, \quad (5)$$

where  $T_k$  represents the  $k$ th value of the authentic values, which are already known to users, and  $N_S$  represents the number of samples. Suppose that there are  $N_S$  samples, the fitness value is written as

$$F(\omega) = \sum_{l=1}^{N_S} E_l, \quad (6)$$

where  $\omega$  represents the vectorization of the  $(\omega_1, \omega_2)$ . Our goal is to minimize this fitness function  $F(\omega)$ , namely, to force the output values of each sample approximate to corresponding target values. At this point, we use our proposed method GPS to optimize formula (6), compared with other algorithms including BP [30], momentum BP (MBP) [31], IGA, and IPS.

**5.2. Simulations and Results.** The datasets consist of T2-weighted MR brain images in axial plane and  $256 \times 256$  in-plane resolution, which were downloaded from the Harvard Medical School website (<http://www.med.harvard.edu/AANLIB/>). We randomly selected 80 images consisting of 40 normal and 40 abnormal. The abnormal brain MR images consist of the following diseases: glioma, meningioma, Alzheimer's disease, Alzheimer's disease plus visual agnosia, Pick's disease, sarcoma, and Huntington's disease. A sample of each is shown in Figure 4.

Figure 5 give the three-level decomposition of 2D DWT decomposition result on a normal brain image. The upper-left corner in Figure 5(b) shows the approximate coefficients serving as the reduced features. The dimension of original image is  $256 \times 256 = 65536$ , while the dimension of approximation image is only  $32 \times 32 = 1024$ .

TABLE 4: Averaged classification accuracy (20 runs).

Algorithm	Classification accuracy
BP	58.813%
MBP	61.687%
IGA	90.063%
IPS	90.625%
GPS	<b>95.188%</b>

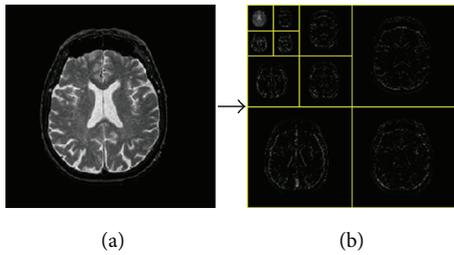


FIGURE 5: A sample of 3-level 2D DWT: (a) a normal MRI brain image; (b) 3-level wavelet coefficients.

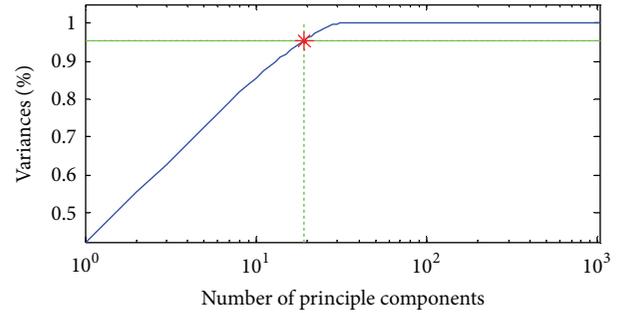
Although the dimensions of extracted features are reduced from 65536 to 1024, this is still a high computation cost. Thus, PCA is used to further reduce the dimensions of features on another level. The curve of cumulative sum of variance with number of principle components is shown in Figure 6. It shows that 19 principle components, which are only 1.855% out of the 1024 features, preserve 95.4% of the total variance.

Those 19 principal components are submitted to the FNN. The proposed GPS method is used to optimize the weights/biases of FNN. Meanwhile, other five methods are also utilized to make a comparison. In order to reduce the randomness, each algorithm was performed 20 times. The averaged classification accuracy of each algorithm is shown in Table 4.

Table 4 indicates that the strategy based on the proposed GPS algorithm obtained higher classification accuracy (95.188%). The IPS and IGA perform nearly the same with classification accuracies about 90%. The MBP and BP algorithms performed the worst with accuracies around 60%. Here, we do not divide the dataset into training and test subsets because we have already employed the  $k$ -fold cross-validation method to avoid overfitting.

## 6. Conclusions and Discussions

Our contributions can be summarized in the following three aspects. First, we proposed a novel algorithm—GPS, for efficient and rapid global search of minimum points. It improved the robustness of pattern search and improved the convergence speed of genetic algorithm. Second, the test and evaluation on five benchmark functions further demonstrate that the proposed GPS is the most robust among the three algorithms. Third, as an example of application, we employed the GPS to the classification of normal and abnormal brain

FIGURE 6: Variances against number of principle components ( $x$ -axis is log scale).

MRI images. The results indicate that GPS is superior to BP, MBP, IGA, and IPS in terms of classification accuracy.

The proposed GPS and the one in [18] are based on the combination of GA and PS. They both make use of the coarse-searching ability of GA and fine-searching ability of PS. Nevertheless, the combining principles are distinct. A question is raised as “which one is better.” Here, we give an assumption that GPS in [18] has a drawback that the PS could start when the GA has not reached the neighborhood of the global optimum yet. GPS in this paper takes the GA as the search method at every step of PS, so it will be more stable but cost more time.

In the future, we will develop simulation experiments to address the question, comparing them in convergence rate, success rate, and computation cost. We will also cooperate with mathematicians to find a theoretical solution. The future work also includes applying GPS to other academic and industrial fields.

## Acknowledgments

The authors express the sincere gratitude to the editor Dr. Vishal Bhatnagar for his careful work and the three anonymous reviewers for their valuable comments.

## References

- [1] G. Corriveau, R. Guilbault, and A. Tahan, “Genetic algorithms and finite element coupling for mechanical optimization,” *Advances in Engineering Software*, vol. 41, no. 3, pp. 422–426, 2010.
- [2] M. Wetter and E. Polak, “Building design optimization using a convergent pattern search algorithm with adaptive precision simulations,” *Energy and Buildings*, vol. 37, no. 6, pp. 603–612, 2005.
- [3] Y. Zhang, L. Wu, and S. Wang, “UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 705238, 9 pages, 2013.
- [4] Y. Zhang, L. Wu, and S. Wang, “Solving two-dimensional HP model for firefly algorithm and simplified energy function,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 398141, 9 pages, 2013.

- [5] L. Moreno, S. Garrido, D. Blanco, and M. L. Muñoz, "Differential evolution solution to the SLAM problem," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 441–450, 2009.
- [6] J. Verboomen, D. Van Hertem, P. H. Schavemaker et al., "Phase shifter coordination for optimal transmission capacity using particle swarm optimization," *Electric Power Systems Research*, vol. 78, no. 9, pp. 1648–1653, 2008.
- [7] Y. Kuroki, G. S. Young, and S. E. Haupt, "UAV navigation by an expert system for contaminant mapping with a genetic algorithm," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4687–4697, 2010.
- [8] I. Kaya, "A genetic algorithm approach to determine the sample size for attribute control charts," *Information Sciences*, vol. 179, no. 10, pp. 1552–1566, 2009.
- [9] S. Kumar and C. S. P. Rao, "Application of ant colony, genetic algorithm and data mining-based techniques for scheduling," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 6, pp. 901–908, 2009.
- [10] M. Z. Jahromi and M. Taheri, "A proposed method for learning rule weights in fuzzy rule-based classification systems," *Fuzzy Sets and Systems*, vol. 159, no. 4, pp. 449–459, 2008.
- [11] A. Jamali, A. Hajiloo, and N. Nariman-zadeh, "Reliability-based robust Pareto design of linear state feedback controllers using a multi-objective uniform-diversity genetic algorithm (MUGA)," *Expert Systems with Applications*, vol. 37, no. 1, pp. 401–413, 2010.
- [12] K. Suga, S. Kato, and K. Hiyama, "Structural analysis of Pareto-optimal solution sets for multi-objective optimization: An application to outer window design problems using Multiple Objective Genetic Algorithms," *Building and Environment*, vol. 45, no. 5, pp. 1144–1152, 2010.
- [13] N. Orlic and S. Loncaric, "Earthquake-explosion discrimination using genetic algorithm-based boosting approach," *Computers and Geosciences*, vol. 36, no. 2, pp. 179–185, 2010.
- [14] C.-W. Tsai, C.-H. Huang, and C.-L. Lin, "Structure-specified IIR filter and control design using real structured genetic algorithm," *Applied Soft Computing Journal*, vol. 9, no. 4, pp. 1285–1295, 2009.
- [15] L. Araujo, H. Zaragoza, J. R. Pérez-Agüera, and J. Pérez-Iglesias, "Structure of morphologically expanded queries: a genetic algorithm approach," *Data and Knowledge Engineering*, vol. 69, no. 3, pp. 279–289, 2010.
- [16] C. Bogani, M. G. Gasparo, and A. Papini, "Generalized pattern search methods for a class of nonsmooth optimization problems with structure," *Journal of Computational and Applied Mathematics*, vol. 229, no. 1, pp. 283–293, 2009.
- [17] X. Song, H. Gu, X. Zhang, and J. Liu, "Pattern search algorithms for nonlinear inversion of high-frequency Rayleigh-wave dispersion curves," *Computers and Geosciences*, vol. 34, no. 6, pp. 611–624, 2008.
- [18] Y. Zhang, L. Wu, Y. Huoc, and S. Wang, "A novel global optimization method- Genetic pattern search," *Applied Mechanics and Materials*, vol. 44–47, pp. 3240–3244, 2011.
- [19] L. De Giovanni and F. Pezzella, "An improved genetic algorithm for the distributed and flexible job-shop scheduling problem," *European Journal of Operational Research*, vol. 200, no. 2, pp. 395–408, 2010.
- [20] T. A. Sriver, J. W. Chrissis, and M. A. Abramson, "Pattern search ranking and selection algorithms for mixed variable simulation-based optimization," *European Journal of Operational Research*, vol. 198, no. 3, pp. 878–890, 2009.
- [21] L. D. S. Coelho, J. G. Sauer, and M. Rudek, "Differential evolution optimization combined with chaotic sequences for image contrast enhancement," *Chaos, Solitons and Fractals*, vol. 42, no. 1, pp. 522–529, 2009.
- [22] R. Ghanbari and N. Mahdavi-Amiri, "Solving bus terminal location problems using evolutionary algorithms," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 991–999, 2011.
- [23] S. Chaplot, L. M. Patnaik, and N. R. Jagannathan, "Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network," *Biomedical Signal Processing and Control*, vol. 1, no. 1, pp. 86–92, 2006.
- [24] E.-S. A. El-Dahshan, T. Hosny, and A.-B. M. Salem, "Hybrid intelligent techniques for MRI brain images classification," *Digital Signal Processing: A Review Journal*, vol. 20, no. 2, pp. 433–441, 2010.
- [25] Y. Zhang, Z. Dong, L. Wu, and S. Wang, "A hybrid method for MRI brain image classification," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10049–10053, 2011.
- [26] X. Delaunay, M. Chabert, V. Charvillat, and G. Morin, "Satellite image compression by post-transforms in the wavelet domain," *Signal Processing*, vol. 90, no. 2, pp. 599–610, 2010.
- [27] J. Camacho, J. Picó, and A. Ferrer, "Data understanding with PCA: structural and variance information plots," *Chemometrics and Intelligent Laboratory Systems*, vol. 100, no. 1, pp. 48–56, 2010.
- [28] R. J. May, H. R. Maier, and G. C. Dandy, "Data splitting for artificial neural networks using SOM-based stratified sampling," *Neural Networks*, vol. 23, no. 2, pp. 283–294, 2010.
- [29] P. Kordík, J. Koutník, J. Drchal, O. Kovářik, M. Čepeck, and M. Šnorek, "Meta-learning approach to neural network optimization," *Neural Networks*, vol. 23, no. 4, pp. 568–582, 2010.
- [30] P. Barmapalexis, F. I. Kanaze, K. Kachrimanis, and E. Georgarakis, "Artificial neural networks in the optimization of a nimodipine controlled release tablet formulation," *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 74, no. 2, pp. 316–323, 2010.
- [31] Z. W. Geem and W. E. Roper, "Energy demand estimation of South Korea using artificial neural network," *Energy Policy*, vol. 37, no. 10, pp. 4049–4054, 2009.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

