INFORMATICA, 2013, Vol. 24, No. 2, 291–313

© 2013 *Vilnius University*

IHPG Algorithm for Efficient Information Fusion in Multi-Sensor Network via Smoothing Parameter Optimization

Wen-Tsai SUNG [*], Ching-Li HSIAO

Department of Electrical Engineering, National Chin-Yi University of Technology
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan
e-mail: songchen@ncut.edu.tw

Received: October 2011; accepted: September 2012

Abstract. This investigate proposed a innovative Improved Hybrid PSO-GA (IHPG) algorithm which it combined the advantages of the PSO algorithm and GA algorithm. The IHPG algorithm uses the velocity and position update rules of the PSO algorithm and the GA algorithm in selection, crossover and mutation thought. This study explores the quality monitoring experiment by three existing neural network approaches to data fusion in wireless sensor module measurements. There are ten sensors deployed in a sensing area, the digital conversion and weight adjustment of the collected data need to be done. This experiment result can improve the accuracy of the estimated data and reduce the randomness of computing by adjustment optimization of smoothing parameter. According to the experimental analysis, the IHPG is better than the single PSO and GA in comparison the various neural network learning model.

Keywords: wireless sensor network, data fusion, improved hybrid PSO-GA, general regression neural network.

1. Introduction

Wireless Sensor Network (WSN) is a new complicated network full of a large number of micro-nodes. Because wireless sensor nods are located in various complex environments, their self-organizing and processing resources are limited and they present serious security problems. Wireless sensor nods belong to complex systems; therefore it's difficult to use a unified model to describe the wireless sensor network nodes. Neural network model can describe the complex systems. Experiments of this study adopts neural network model to describe the wireless sensor nodes and then improve problems of data distortion and data redundancy via Improved Hybrid PSO-GA (IHPG) algorithm based on adjustment optimization of smoothing parameter. Wireless sensor network is a task-oriented wireless network composed of numerous nods. It integrates many areas of technology, such as sensor technology, embedded computing technology, modern Internet, wireless communication technology and distributed information processing technology,

[*] Corresponding author.

then through various micro-sensors it carries out real-time monitoring on the target information, through embedded computing resources it carries out processing of information, and through wireless communication network it sends information to remote users (Yong Fang, 2000). This technology has very promising applications in many fields, including military defense, industrial and agricultural control, urban management, biomedical field, environmental monitoring, disaster relief, remote control of dangerous areas, and presents research values and practical values.

In addition, compared with the wired communication, in terms of reliability, security and scalability, wireless communication technology presents advantages. Based on the above considerations, it appears that wireless data acquisition system should be built (Zurada, 1992). Existing wireless data acquisition systems often use a single wireless transmission mode, thus they present some weak points such as poor stability and lack of accuracy of data acquisition. This study presents some experiments neural network model construction method for verifying the advantages of IHPG algorithm in this figure 1 system; wireless communication technology achieves the transmission of specific collected data, ultimately ensures that the system can flexibly and accurately fulfill wireless data acquisition and transmission, and ultimately improves the anti-jamming capability of this data acquisition system. There are rapid changes in environmental temperature on the ordinary industrial measurement and control site.

Most sensors have certain sensitivity to temperature, therefore the rapid changes of temperature will make the sensors' zero point and sensitivity start to change, sensed data will vary with environmental temperature change, and then additional errors will occur. Therefore, temperature compensation has been a crucial element of the industrial measurement and control system. This study adopts wireless method to discuss the transmission data in sensor networks, and this method allows to flexibly collecting data from outdoors, but when the sensors collect data, there will be some problems that result in data distortion, such as power shortage, sensor's aging and sensor's location. The errors in collected data will affect the operation of the plant or cause workplace accidents. Therefore, if the status of plant equipment can be controlled clearly, the problems could be early identified and possible dangers could be avoided. This study adopts neuron to describe the network nodes in the wireless sensor network system, uses neuron mathematical model to represent wireless sensor network system, and applies neural network to data fusion in wireless sensor network, thus flexibly and accurately carries out wireless data acquisition and transmission with less redundant data as well as lower cost, and improves the anti-interference ability and efficiency of the data acquisition system. Finally, in the calculation of data fusion this study proposes IHPG algorithm to carries out analysis and discussion with examples of the actual simulations based on Artificial Neural Networks (ANN), Radial Basis Function (RBF) and General Regression Neural Network (GRNN) neural network integration algorithm.

2. Neural Network Approach to Data Fusion

In order to obtain complete information with more reliable decision and less uncertainty, it has become an important research topic to propose, among many studies, an information fusion technology to the integrate a large amount of measurement results obtained from multiple information sources. Many scholars have studied how to combine predictions coming from multiple classifiers to generate a single classifier with all the effects. In 1996 according to the statistics Breiman concluded that single classifier presented an unstable nature (Bhattacharyya *et al.*, 2001). This is because that the single classifier might present inferior classification performance in certain components of training data, thus it affects considerably the results of classification. Compared with a stable classifier, an unstable classifier is characterized by its small deviation but great variation. So it is expected that combining multiple classifier could reduce the deviation and variation. Wireless Sensor Network is a new complicated network full of a large number of micronodes. Because wireless sensor nods are located in various complex environments, their self-organizing and processing resources are limited and they present serious security problems. Wireless sensor nods belong to complex systems; therefore it's difficult to use a unified model to describe the wireless sensor network nodes. Neural network model can describe the complex systems. This study adopts neural network model to describe the wireless sensor nodes and then improve problems of data distortion and data redundancy.

For some complex monitored objects, it often needs a greater number of sensors to capture various forms of monitored information in order to make more accurate judgments about the status of the monitored objects. The mine safety monitoring, for example, must be always detecting the information about gas, mineral dust, fire and flood. Therefore, it is necessary to design a reliable safety monitoring system in order to achieve to ensure security. Besides, compared with the wired communication, in terms of reliability, security and scalability, wireless communication technology presents advantages. Based on the above considerations, it appears that wireless data acquisition system should be built. Existing wireless data acquisition systems often use a single wireless transmission mode, thus they present some weak points such as poor stability and lack of accuracy of data acquisition. This study employed IHPG algorithm to general neural network model for information fusion in this WSN system; wireless communication technology achieves the transmission of specific collected data, ultimately ensures that the system can flexibly and accurately fulfill wireless data acquisition and transmission, and ultimately improves the anti-jamming capability of this data acquisition system. This investigate describes the data fusion model design based on smoothing parameter optimization technology via neural network model carries out simulation and evaluation of this model (Chair *et al.*, 1986). According to the characteristics of specific applications, using neural network in WSN that presents some advantages: (1) that data fusion point is infinitely close to the data source node will help maximize the performance of data fusion; (2) neural network algorithm can abstract the data well and draw the important features of data; (3) neural network algorithm provides a diversity that is well adapted to the needs of different applications.

Neural network approach can exhibit properties analogous to adaptive biological learning; it has good sensors signals transmission capabilities, and learning, information recall resistant to hardware or data fusion. It has the following advantages over the statistical approaches: distribution-free; and degree of belief in each data source is represented by the weights of the network and determined by the training process. It is not necessary to estimate the reliability function during the fusion process. Bowman and Priebe and Marchette suggest that neural networks are superior to traditional cluster methods for identity fusion, especially when the input data are noisy and when data are missing. However, the theoretical basis of neural networks is still evolving, during the implementation of a neural network, the problem of local extremum, convergence speed of the training, and misclassification when the data dimensions increase still should be considered (Chiuderi *et al.*, 1994).

3. The Structure and Operating Procedures of the System

Wireless sensor network usually consists of sensor nodes, sink nodes and management nodes. The hardware architecture of this study is shown in Fig. 1. A large number of sensor nodes are randomly deployed within or near the monitored region and they are able to form a self-organized network. The monitored data coming from the sensor nodes will be transferred along the other sensor nodes hop by hop. In the transmission process the monitored data may be handled by multiple nodes, through a multiple hops they will go to the sink nodes, and then finally through the Internet or satelite they will arrive in the management nodes. Through the management nodes the user carries out the configuration and management of the sensor network, launches monitoring tasks and collects monitored data. The sensor nodes are usually densely deployed in the monitored region. Any event that occurs in this monitored region might be monitored by multiple nodes, then these detected data will be transferred to the outside world, and that will result in an unnecessary waste of energy. The sensor nodes could be adjusted in order to make only some nodes keep working but the others stay dormant, and make all events that occur in the monitored region can be detected by the working nodes. The resting nodes are in a low-power state and are ready to wake up periodically so that if any active node fails to work due to an energy problem or other reasons, the resting nodes can replace any disabled node. Through a rotation of the work by the sensor nodes, we can reduce the energy consumption due to the event detections, communication conflicts, and transmissions of redundant data. That

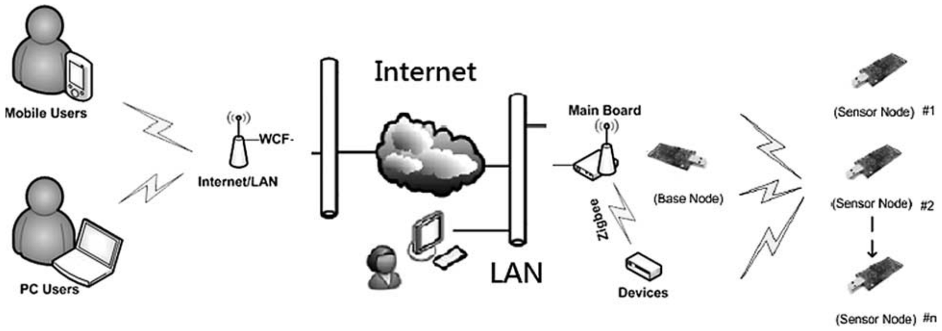


Fig. 1. The structure of the sensor network system.

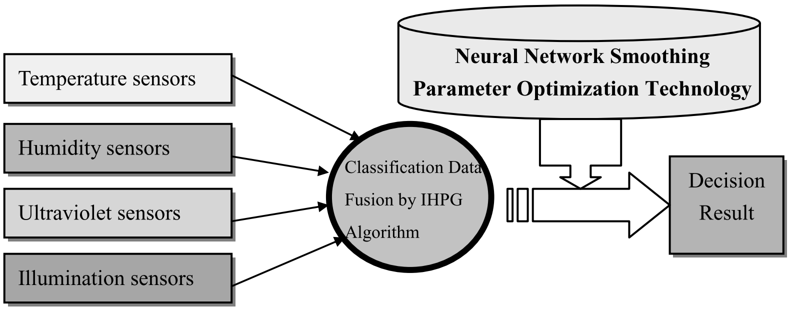

Temperature sensors
Humidity sensors
Ultraviolet sensors
Illumination sensors
Classification Data Fusion by IHPG Algorithm
Neural Network Smoothing Parameter Optimization Technology
Decision Result

Fig. 2. System data fusion and multi-sensor network framework.

could effectively extend the lifetime of the system (Bowden *et al.*, 2006; Dayhoff, 1990; Specht, 1991; Sung and Tsai *et al.*, 2011a).

This data fusion system framework is shown in Fig. 2. Classification fusion technology is applied to real-time signal recognition from multiple sensors data in a wireless sensor network with a node-sink mobile network structure. These wireless sensors include temperature, humidity, ultraviolet and illumination in four variable measurements for a storehouse boundary fire detection warning system.

The target of sensor node scheduling algorithm is to make a smallest number of active nodes keep working, and have any event that occurs in the monitored region be detected by any active sensor node. It is possible that the initial sensor nodes can’t completely cover the monitored area, so the active sensor nodes are generally required to completely cover the initially covered area. Thus, if any event that occurs in this monitored area can be detected by any node, this event can definitely be detected by any active sensor.

Figures 3(a)–(d) are the hardware components designed respectively for this study. The type of the micro-controller chips on the coordinator’s circuit board is CC2430F128. The circuit board can be directly powered by a 9V DC power adapter or by USB. The coordinator links through USB via COM port interface to the computer. And the computer collects, through the coordinator, the information from every node and controls these nodes (Zhang *et al.*, 2004) The type of the micro-controller chips on the circuit board of target node is CC2430F128. In the positioning applications it is used as reference node of target node. In the positioning applications it needs at least 3 target nodes to position correctly. It can use at most 16 target nodes. In theory, more nodes will result in a higher accuracy. The target node can also be used as sensor acquisition in the environmental applications. This study uses also this component to collect the detected data. As to the operation of the wireless temperature sensor module, after writing an application programming CC2430 the temperature could be read and it will control UART and transfer the temperature to the coordinator. The simulator is used to burn the firmware programming on to the circuit board of the target node/node to be positioned in order to carry out the debug test and formal execution of the application programming. The data fusion method based on the neural network in the multi-sensor applications helps to develop the data fusion method based on the neural network in the wireless sensor appclications proposed by this study (Sung *et al.*, 2010a). This study proposes an intelligent approach of multi-sensor

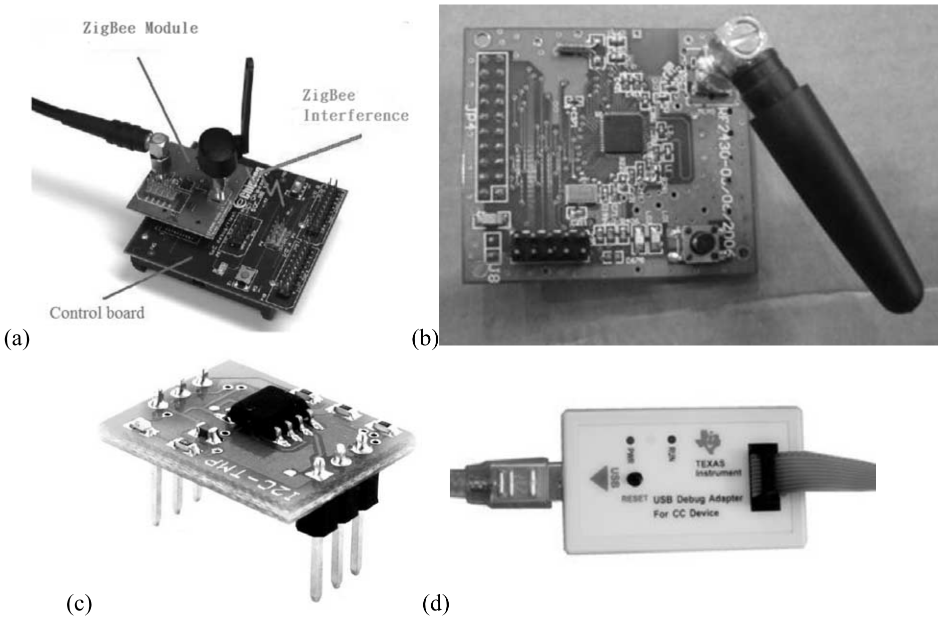


Fig. 3. (a) Coordinator's circuit board, (b) node-set, (c) wireless temperature sensor module, (d) T1 CC2430/31 simulator.

data based on the neural network, which makes good use of the nonlinear mapping ability of the neural network to intelligently deal with the detected data coming from different sensors simultaneously by identifying accurately the various signals, reducing effectively the errors and false alarm rate of the sensors, and enhancing the anti-jamming capability and adaptability to the environment of the system (Sung and Tsai *et al.*, 2011b).

4. Classification Data Fusion by IHPG Algorithm

Existing literatures combined the advantages of genetic algorithm and PSO algorithm, proposed a hybrid algorithm of genetic algorithm and PSO algorithm, but in the early calculus, stochastic too much is a fatal flaw. PSO is a class of biologically inspired heuristics optimization algorithms and swarm intelligence-based modeling technique which was developed by Kennedy and Eberhart (Wu *et al.*, 2012). Several researchers have combined PSO and GA to form different hybrid algorithms. Shi *et al.* (2003) combined a variable population-size genetic algorithm (VPGA) and PSO to form a hybrid algorithm that generates initial populations for VPGA and PSO according to a certain proportion. The new population is obtained by evolution according to the corresponding algorithm rules. Gandelli *et al.* randomly divided the total population into two subpopulations and evolved the two subpopulations using GA and PSO operations, respectively. Kao and Zahara (2008) proposed a GA-PSO hybrid algorithm for geometric proportional populations. In this method, an Unler developed a model, using a PSO-based energy demand forecasting, to forecast the energy demand of Turkey more efficiently. Further, to avoid the particle to be stuck in the local minimum, Kuo and Han (2011) integrated the mutation mechanism of GA with PSO. Valdez *et al.* combined GA and PSO using fuzzy logic to integrate the results of both methods and for parameter tuning. Particle swarm optimization algorithm improvements have been done a lot of research, also produced a hybrid algorithm of particle swarm optimization algorithms and genetic algorithms. These algorithms are initially allocated to each particle a mixture of probability, and then mixing probability to select a certain number of particles in each iteration according to a mating pool, the pool of particles with random pairs, through crossover operator to produce two sub-individual, sub-individual to replace the parent individual in order to maintain the populations of the same number of each parent individual (Shahram *et al.*, 2009).

The PSO-GA algorithm has been improved by Shi *et al.* (2003) and Matthew and Terence (2005). PSO-GA hybrid algorithm is the use of the randomness of the genetic algorithm to increase the search range, and then use the particle swarm algorithm for a more detailed search around to find the optimal particle. From the experiments in this study to analyze the performance of the PSO-GA is superior to individual genetic and particle swarm algorithm. Improved Hybrid PSO-GA (IHPG) algorithm be proposed in this study are to improve on how to reduce the randomness, Suppose a group consisting of m D-dimensional particle swarm,The ith particle represent by a D-dimensional vector Xi $(i = l, 2, \dots, m)$, it shows that the location of the particles in the search space for the $X_i(X_{i1}, X_{i2}, \dots, X_{iD})$ $(i = l, 2, \dots, m)$. The location of each particle is a potential solution. Through this position on the specific objective function be calculated the fitness value of the particles. If it is a higher fitness value, the corresponding particles is well. The first particle's "flying" rate is also a D-dimensional vector, expressed as $Vi(V_{i1}, V_{i2}, \dots, V_{iD})$ $(i = 1, 2, \dots, m)$. The first particle history of the best location in mind for $Pi(P_{i1}, P_{i2}, \dots, P_{iD})$. The best particle position in the group denoted as $P_g(P_{g1}, P_{g2}, \dots, P_{gD})$. The position and speed of the standard PSO update formula (1) and (2):

$$v_{id}(t) = v_{id}(t-1) + c_1 r_1 \big(p_{id} - x_{id}(t-1)\big) + c_2 r_2 \big(p_{gd} - x_{id}(t-1)\big), \tag{1}$$

$$x_{id}(t) = x_{id}(t-1) + v_{id}(t). \tag{2}$$

This study proposed IHPG algorithm that velocity and position be updated via formula (3) and (4) to define:

$$\begin{aligned} v_{id}(t) = X\Big(&wv_{id}(t-1) - c_1 \,\mathrm{Rand}_1\big(p_{id} - x_{id}(t-1)\big) \\ &+ c_2 \,\mathrm{Rand}_2\big(p_{gd} - x_{id}(t-1)\big)\Big), \end{aligned} \tag{3}$$

$$x_{id}(t) = x_{id}(t-1) + v_{id}(t), \tag{4}$$

$$\kappa = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \tag{5}$$

where $\varphi = c_1 + c_2,\ \varphi > 4$, X_{id} is particle i's position at its d-dimensional; V_{id} is particle i's velocity at its d-dimensional; w is inertia weight; acceleration constant c_1, c_2 are two non-negative, $\mathrm{Rand}_1,\ \mathrm{Rand}_2$ are uniformly distributed random number between $[0,1]$. P_{id} is particle i in the d-dimensional individual historical best position; P_{gd} is historical best position of all particles in the d-dimensional. In IHPG algorithm, φ is set 4.1 and the X become 0.73. According to this study recommended, w is set to 0.8 and linearly decreasing to 0, the maximum rate is Vmax and be set to the Xmax (Pohl and Genderen, 1998).

Standard GA will become the solution of the problem encoding the chromosome, and then define an initial population of individuals in the population represents part of the problem feasible solution, the solution space of the search space is composed by different chromosomes. Before the start of the search, the search space randomly be selected series of chromosome formation of the initial population, and then calculate the individual to adapt to the specific objective function value, competitively select individuals. In turn use such as selection, mutation and crossover genetic manipulation to a new generation of chromosomes of high fitness value than the previous generation of chromosome. This process of iteration until termination condition is met; the best chromosome of the last generation is the final solution.

Initialization the largest population of N, and calculate the individual fitness value; when the conditions are not met then the stopping rule, first of all individuals on the parent population in the sort order $(N/2)$ according to adapt to the sort of value size; individual parent individuals according to the probability of crossover operation, if the child individual is better than the parent individual to retain the sub-individual, otherwise leave the parent individual; probability of random mutation operation, if better than the original individual is to keep the new individual, otherwise retain the original individual; steps to re-calculate the fitness value; each crossover and mutation, and the parent individuals, the original individual, just out of structurally similar to the parent of the individual and the former individual so as to maintain the diversity of the population. The crossover operation using particle position dimension crosses, the Crossover probability is $\mathrm{Crossover}_{\mathrm{ratio}}$, shown in Fig. 4.

Mutation probability is $\mathrm{Mutatuon}_{\mathrm{ratio}}$, the mutation operation may produce three kinds of results:

(1) 90% probability: $\mathrm{gene}_{\mathrm{mutated}} = \mathrm{gene}_{\mathrm{cumrnt}} + X_{\max} \times \varphi/3$,

(2) 12% probability: $\mathrm{gene}_{\mathrm{mutated}} = \mathrm{gene}_{\mathrm{cumrnt}} \times (M_A)^{\varphi}$, which M_A is variation to change the parameters. φ is uniform random number, the range is $[0.0, 0.1]$.

(3) 8% probability: $\mathrm{gene}_{\mathrm{mutated}} = \mathrm{gene}_{\mathrm{cumrnt}} \times \rho$, which ρ is a uniform random variable and its the range is $[-1,1]$.

IHPG algorithm steps are as follows: (1) initialize the number is the particle swarm of the popsize; (2) calculate the fitness of the particles and retain optimal particle position

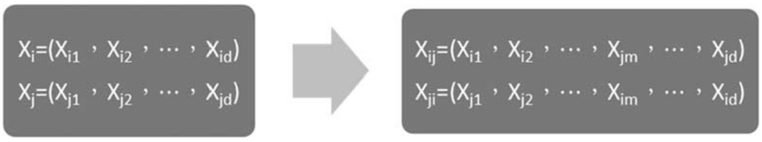

$X_i=(X_{i1}, X_{i2}, \cdots, X_{id})$
$X_j=(X_{j1}, X_{j2}, \cdots, X_{jd})$
$X_{ij}=(X_{i1}, X_{i2}, \cdots, X_{jm}, \cdots, X_{jd})$
$X_{ji}=(X_{j1}, X_{j2}, \cdots, X_{im}, \cdots, X_{id})$

Fig. 4. Crossover operating diagram.

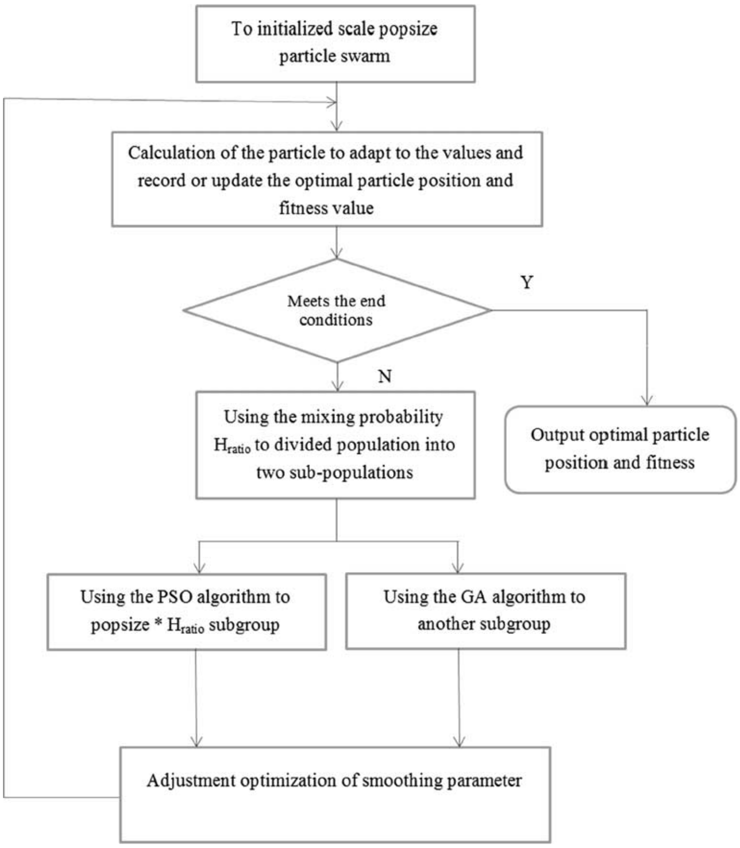

To initialized scale popsize particle swarm
Calculation of the particle to adapt to the values and record or update the optimal particle position and fitness value
Meets the end conditions
Y
N
Using the mixing probability H_{ratio} to divided population into two sub-populations
Output optimal particle position and fitness
Using the PSO algorithm to popsize * H_{ratio} subgroup
Using the GA algorithm to another subgroup
Adjustment optimization of smoothing parameter

Fig. 5. IHPG algorithm flowcharts.

and fitness value; (3) using the mixing probability H_{ratio} to divided the particle swarm into two subgroups; (4) using the PSO algorithm to popsize $\times H_{\text{ratio}}$ subgroup; (5) using the GA algorithm to the remaining subgroup; (6) adjustment optimization of smoothing parameter; (7) repeat Steps (2)–(6) until the computing result meet the closing conditions. Algorithm flow chart is shown in Fig. 5 (Viswanathan and Varshney, 1997; Welstead, 1994; Xing *et al.*, 2011).

This study proposed the IHPG algorithm, which could optimize the coefficients of equations with better performance. The main differences of our approach from existing approaches can be summarized in the following three aspects (Zhang *et al.*, 2004; Zurada, 1992):

1. To better optimize the coefficients, an effective IHPG optimization algorithm was developed based on PSO and GA, which can fully combine the merits of these two methods without their drawbacks.
2. Based on the full analysis of the sensors demand-affecting factors, five factors, which are different from the WSN indicators in existing research, are chosen as the inputs of our model.

3. Reduce the randomness of computing by adjustment optimization of smoothing parameter.

5. Optimization of Smoothing Parameter

For general neural network, if the learning samples are determined, the link weights between the corresponding network structures and neurons will be also determined, and the network learning is actually a determination processing of the smoothing parameter σ. There are basically two ways of selection of the smoothing parameter. One is based on the experience, through the spreadsheet, comparison, overall consideration, to fit and predict the accuracy, and to select the suitable value. But this method there is, after all, man-made and subjective factors, it often can not obtain the optimal value of σ. The other is to build a suitable objective function by the method of optimization. Seng had proposed an algorithm similar to the BP gradient descent algorithm to find σ, but this method is easy to fall into a local minimum (Seng *et al.*, 2002). A reasonable selection of the smoothing parameter is an important thing. This study uses the global search ability of genetic algorithm for the optimization of σ. Genetic algorithm is an analog of genetic selection and biological evolution of the natural world. According to the intended target adaptability, the function carries out individual assessments, based on the evolutionary mechanisms of survival of the fittest; it can obtain a better population and finally achieve the best result that meets well the requirements. It consists of five elements: parameter encoding, initial population settings, fitness function design, genetic operation design and control parameters design. Among them, the selection of control parameter crossover probability P_c and mutation probability P_m is the key affect on the behavior and performance of genetic algorithm and it has a direct effect on the population diversity and the algorithm convergence. This study adopts Adaptive GA (AGA) method where P_c and P_m are able to change automatically with adaptability. When the adaptability of each individual population tends to consistency or local optimum, P_c and P_m will increase; when the adaptability of each individual population tends to dispersion, P_c and P_m will decrease. Meanwhile, for the individual whose adaptability is higher than the average adaptability of the population, it will be corresponded by lower P_c and P_m in order to protect the result and make it enter to the next generation; for the individual whose adaptability is lower than the average adaptability of the population, it will be corresponded by higher P_c and P_m in order to eliminate the result. The operations are as follows:

$$P_c = \begin{cases} \frac{k_1(f_{\max}-f')}{f_{\max}-f_{\text{avg}}}, & f' \geqslant f_{\text{avg}}, \\ k_2, & f' < f_{\text{avg}}, \end{cases} \tag{6}$$

$$P_m = \begin{cases} \frac{k_3(f_{\max}-f)}{f_{\max}-f_{\text{avg}}}, & f \geqslant f_{\text{avg}}, \\ k_4, & f < f_{\text{avg},} \end{cases} \tag{7}$$

where $f_{\max}$ is the highest adaptability between the population f_{avg} is the average adaptability of every population f' is the higher adaptability between the two individuals to

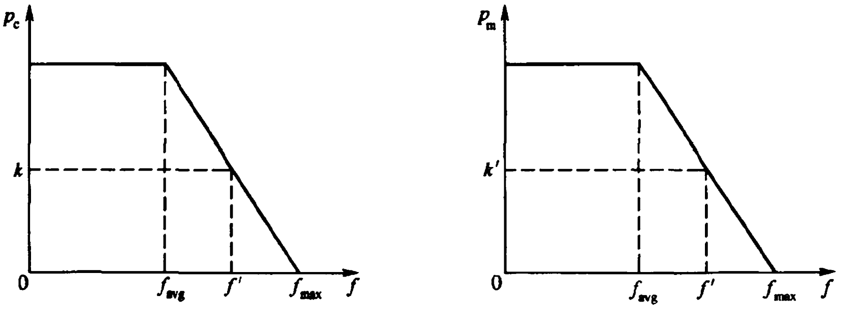


Fig. 6. Adaptive Genetic Algorithm.

cross f' is the mutation individual's adaptability k_1, k_2, k_3, k_4 takes the value between $[0, 1]$. The analysis of (6) and (7) demonstrates: the relationship between adaptability, crossover and mutation rates presents a simple linear mapping, shown in Fig. 6.

$$k_1 = k_2 = k, \qquad k_3 = k_4 = k'.$$

In the analysis Fig. 6, we can find that when the adaptability is getting closer to the maximum adaptability, crossover and mutation rate are getting smaller; when the adaptability is equal to the maximum adaptability, crossover and mutation rate are 0. This adjustment method is more suitable for the later stage of the group evolution but is unfavorable for the early stage of the evolution. This is because that in the later stage of evolution, each individual in the group is basically able to show a better performance, and at this stage, it's not suitable to change the individuals a lot in order to avoid destroying the individual's excellent performance structure. And in the early stage, the better individuals in the group stay almost in a constant state, and these excellent individuals are not verified to be the optimal solution in the overall situation. Therefore, we try to improve this method by making the crossover and mutation rate of the individul with the highest adaptability in the group not be 0, and increasing respectively P_c and P_m. Through the above improvement, the calculation expression of P_c and P_m as follows:

$$P_c = \begin{cases} p_{c1} - \frac{(p_{c1} - p_{c2})(f' - f_{\text{avg}})}{f_{\max} - f_{\text{avg}}}, & f' \geqslant f_{\text{avg}}, \\ p_{c1}, & f' < f_{\text{avg}}, \end{cases} \tag{8}$$

$$P_m = \begin{cases} p_{m1} - \frac{(p_{m1} - p_{m2})(f_{\max} - f)}{f_{\max} - f_{\text{avg}}}, & f \geqslant f_{\text{avg}}, \\ p_{m1}, & f < f_{\text{avg}}. \end{cases} \tag{9}$$

P_{c1} takes 0.7– 0.9, P_{c2} takes 0.5–0.6, P_{m1} takes 0.1–0.2, P_{m2} takes 0.005–0.06, the application of genetic algorithm in this study is based on it. In the application,the parameters P_{c1}, P_{c2}, P_{m1} and P_{m2} all take the maximum within their own range. The specific process is as follows: (1) adopt real number encoding, the range is $[0.001, 6]$; (2) the initial smoothing parameter group is created by random generation method, the scale is 60; (3) the select operand adopts roulette method and the best retention policy, cross operand adopts arithmetic crossover, mutation operand adopts non-uniform mutation, crossover and mutation probability adaptively; (4) termination algebra is 350. Adaptability function is :

$$\mathrm{fit}_{\mathrm{GNN}} = \frac{1}{\frac{\alpha}{n_1}\sum_{i=1}^{n_1}(y_i - \hat{y}_i)^2 + \frac{\beta}{n_2}\sum_{j=1}^{n_2}(y_j - y_j)^{2\hat{}} + 1}, \tag{10}$$

where y_i is the measured output of the top 500 training data, $\hat{y}_i$ is the predicted output of the training data (the determined function in the Step (2)), n_1 is the number of training samples; y_j is the measured output of the last 500 training data, $\hat{y}_j$ is the predicted output of the training data, n_2 is the number of verified samples; α and β are the weight coefficients, here we take $\alpha = \beta = 0.5$. Here we put in the testing samples is to improve the network's promotion ability.

6. Simulation Experiment in General Neural Network

(1) Artificial Neural Networks (ANN)

Because of development of Artificial Neural Networks (ANN) technology and advancement of computer algorithms, ANN has developed many interdisciplinary applications from its program control functions, including system identification and prediction, classification, parallel simulator, fault diagnosis, character recognition, data compression, and neural network controller. The environmental applications of these functions have been developed in recent years. This is mainly because that in front of some environmental factors such as chaotic time series and nonlinear behaviors ANN has still very good predictive ability due to sufficient data. ANN can be considered as complex network composed of a large number of simply connected artificial neurons in order to be able to mimic the ability of biological neural networks. The relationship between the input value and output value of the artificial neurons can be shown as follows (Zurada, 1992):

$$y_j = f\Big(\sum w_{ji}x_i + \theta_j\Big), \tag{11}$$

where $y_j =$ the output signal (i.e., output variable) of mimicry of the biological neuron No. j; $f =$ transfer function of the mimicry of biological neuron model; $w_{ji} =$ synaptic strength (i.e., link-weighted value) of the mimicry between the biological neuron No. i and No. j; $x_i =$ the input signal (i.e., input variable) of mimicry of the biological neuron No. i; $\theta_j =$ the threshold (i.e., threshold limit value) of mimicry of the biological neuron No. j.

As to the construction of ANN, it can be divided into two stages including learning and recall. In the learning stage, mainly, the sample data are used in accordance with its learning rules in order to change and adjust the link-weighted values between neurons. And, the learning algorithm can be divided into supervised one or unsupervised one. The supervised one is generally applicable to the problems of function approximation, but the unsupervised one is often applied to solve the problems of data classification.

In the recall stage, the output of the network is calculated based on the input data. From the above information we know that ANN is composed of many nonlinear mathematical operation units and multiple links between these operation units. The operation way of these operation units is usually parallel and distributed so that it can deal with a large number of data and that's why it is applied to various applications that need to deal with a large number of data operations. Therefore ANN's special advantage is that it does not need to know what the system's mathematical model is, it uses the neural network to replace the system's model directly, and it can also obtain the relationship between the input and output. Therefore, the network learning process, even the above-mentioned minimization process of an energy function, the method of steepest descent is usually used to correct it, that is, every time when entering a training example, the network will slightly adjust the weight, the adjustment extent and the error function are propotional to the sensitivity with respect to the weight, that is, they are propotional to the partial derivative of error function with respect to the weight. As follows:

$$\Delta W_{ij} = -\eta \cdot \frac{\partial E}{\partial W_{ij}}, \tag{12}$$

where W_{ij} is the link weight between the processing unit No. i of the layer No. $n - 15$ and the processing unit No. j of the layer No. n. And, η is the learning rate that uses the method of steepest descent to control and minimize the stride of the error function every time. $\partial E/\partial W_{ij}$ can be obtained by using the chain rule in calculus. This learning process is usually carried out by one training example for each time. As all training examples are learned, it is called a learning cycle. A network can take several learning cycles to learn the training examples until it achieves the convergence (Fausset, 1994; Sung *et al.*, 2010b).

(2) Radial Basis Function (RBF)

The performance of RBF network model depends on the type, number and center of the selected radial basis function, as wall as the algorithms of the network link weights. Generally speaking, the radial basis function network model built by this study can be divided into three steps: (1) determine the radial basis function's center cj; (2) calculate the radial basis function's width σj; (3) obtain network link weight wkj. As to the selection of the function's center of the radial basis function network model, some researchers randomly select from the training examples, some select it by using unsupervised or supervised methods. In the process, the input layer is used to accept the information coming from the outside world, through the treatment by the neurons in the hidden layer step by step the result will be transfer to the output layer; if the output layer can't obtain the expected output values it will use the error feedback message to correct every neuron's link weights and thresholds to reduce the error values. The above-mentioned error can be obtained by the training example's expected output value and the network's deduced output value. But generally people use the error cost function E as an important indicator of the assessment of network learning quality, shown as follows (13):

$$E = \frac{1}{2}\sum_{k=1}^{n}(t_k - y_k)^2, \tag{13}$$

where $t_k =$ expected output value of the neuron No. k in the output layer in the training example; $y_k =$ deduced output value of the neuron No. k in the output layer in the training example. From (13), we know that the error cost function is a quadratic function, thus we can use Calculus to find extrema to find a set of best parameters to make (13) obtain the minimum (Jagyasi *et al.*, 2006). There are many methods that can make the error cost function achieve the minimum, and the common method is steepest descent. That is, in every training example, we find the partial derivatives of the error cost function $E(n)$ before the learning with respect to w_j, c_j and σ_j in order to achieve the correction of each parameter $\Delta w_j(n),\ \Delta c_j(n)$ and $\Delta\sigma_j(n)$, and obtain each parameter after learning (i.e., in the next training). Besides, we can adopt the momentum term to improve the convergence rate of neural network. That is, when we calculate the change values in weights, we can add some weight in order to consider the impact that the change value $\Delta w_j(n-1)$ in the last phase has on this phase. The equation is as follows:

$$\Delta w_j(n) = -\eta\frac{\partial E(n)}{\partial w_j} + \alpha\Delta w_j(n-1), \tag{14}$$

where $\alpha =$ momentum factor, $\eta =$ learning rate. Generally speaking, a smaller η will need more learning cycles; a larger η can reduce effectively the number of learning cycles, but it might enlarge the magnitude of the modification of the weights then exceed the minimum space point of the desired error function. Therefore, in the learning it needs to use the trial-and-error method to determine η.

To sum up, before the learning the radial basis function network model must firstly set the learning rate, the initial link weights, the maximum allowable error, the inertia factor and the maximum number of learning cycles. And, the training process of the radial basis function network model is to use error's reverse transmission and appropriate supervised learning rule, such as the equation (14) based on the method of steepest descent, to constantly modify the network parameters such as the weighted value, and to make the network's deduced output value get closer to the expected output value. And if any condition below is achieved: (1) the overall value of the error cost function achieves an acceptable convergence; (2) the number of trainings achieves the maximum number of learning cycles; (3) the error gradient is less than the settings, the network will terminate the learning (Srinivas *et al.*, 1994).

(3) General Regression Neural Network (GRNN)

GRNN is a type of supervised learning network. This network belongs to the supervised learning and its basic concept of principle was inspired from the probability model. However, the probabilistic neural network is only suitable to be used for classification and it is not able to solve the problems of continuous variables. Thus in 1991, Donald F. Specht had proposed the learning algorithm of General regression neural network.

GRNN was evolved from the probabilistic neural network. This algorithm not only can deal with the problems of classification, but also is used to learn dynamic mode and predict or control. And, it also presents a good capacity to deal with the problems of leaner or nonleaner regression. GRNN was evolved from the probabilistic neural network, and the probabilistic neural network was inspired from the probability model. Therefore, the basic concept of the principle of GRNN was extended from the probability model. A group of training samples with M-dimensional input vector can be considered as the sample points in the M-dimensional space. GRNN uses these sample points to estimate an unknown sample point function value. Because GRNN was inspired from the probability model, it doesn't need to assume a specific functional form in advance as the traditional regression analysis, and it only needs to be presented by the form of probability density function (Sung and Tsai *et al.*, 2011b).

In the regression equation between dependent variable Y and the independent variable X, Y usually represents the system's output value and X represents the input value. GRNN doesn't need to assume a specific functional form in advance as the traditional regression analysis, and it only needs to be presented by the form of probability density function. Suppose that $f(x, y)$ is the joint probability density function of variable X and variable Y. x is a measurement of variable X. Then the regression equation between Y and x is:

$$E(Y|X) = \frac{\int_{-\infty}^{\infty} yf(x,y)\,dy}{\int_{-\infty}^{\infty} f(x,y)\,dy}. \tag{15}$$

But $f(x, y)$ is unknown, we have to use the observed values of X and Y to estimate $f(x, y)$. Here we use Parzen's windows nonparametric method to estimate $f(x, y)$, and s is the smoothing parameter, a constant greater than 0; the smoothing parameter is the only parameter that needs to be determined by learning method in GRNN. If we simplify (15), we obtain:

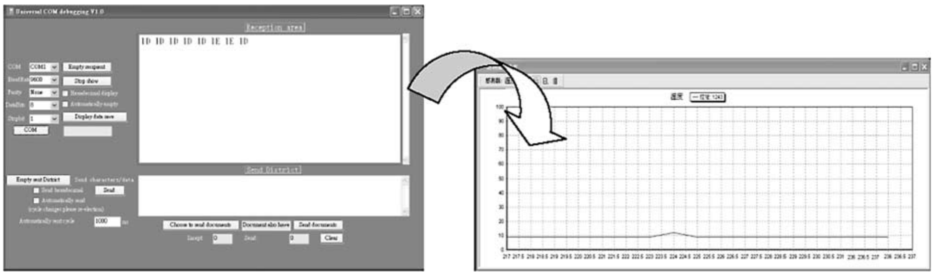
$$\tilde{y}(x) = \frac{\sum_{i=1}^{n} y^i \exp\left[-\frac{(x-x^i)^T(x-x^i)}{2\sigma^2}\right]}{\sum_{i=1}^{n} \exp\left[-\frac{(-x^i)^T(x-x^i)}{2\sigma^2}\right]} = \frac{\sum_{i=1}^{n} y^i \exp\left[-\frac{D_i^2}{2\sigma^2}\right]}{\sum_{i=1}^{n} \exp\left[-\frac{D_i^2}{2\sigma^2}\right]}, \tag{16}$$

where x_i and y_i are the sample values of variable X and variable Y; D_i^2 is $(x - x^i)^T \times (x - x^i)$, the estimated value $\tilde{y}(x)$ is the weighted average of the all samples' observed value y^i, σ is the smoothing parameter. When a new vector X enters the network, this vector minus the training example's vector, the squared value of the difference between the two vectors will be added to the sum and input to the nonlinear activity function, the result of this activity function is the output value of the type unit. And, the output value of the type unit will be sent to the summation unit. GRNN's advantage is that when the number of the training samples is large, it can learn rapidly and converge to the best regression plane with maximum accumulated samples, and it can deal with the unstable data (Bhattacharyya *et al.*, 2001; Blum *et al.*, 1997).

7. Experiment Analysis and Result Discussion

The temperature data collected by wireless sensor nodes are shown as the inferface in Fig. 7. When the temperature data are collected, the collected data are hexadecimal data and that will be converted to decimal, for example, if a collected hexadecimal data is 1D, that will be converted to 29, in order to obtain a more accurate result. The data format must be able to use as the neural network input vector data and the actual output vector data. The testing data is regarded as cost study; the neural network toolbox in MATLAB is used as neural network training interface.

At first, we build in MATLAB work space the variables of the target value of training and target output value. The target value is set between -10 degrees to 40 degrees Celsius. The collection is carried out every 10 seconds. A collection of 400 data is the the target value. We make good use of the newrbe function. The newrbe function, offered by the toolbox in MATLAB, can generate a network that presents no error with respect to the determined training sample vectors. The only condition is that an enough large spread constant should be offered in order to make neurons input range be able to cover an enough large area. In this way, any input at any time will allow some radial neurons to generate great response output. This way will make the network transfer function become smoother and can better reflect generally the new input vectors that are different from the sample vectors. In view of accuracy and enough training time, we adopt a neural network with 15 neurons as testing model. The number of network trainings (epoch) and the error convergence are shown as from Figs. 8(a) to 8(c). As shown in Fig. 8(a), from the network error convergence we can find that the training network with 15 neurons has 400 training targets and 1000 training samples; about with 5 training cycles the errors will be reduced but the result is still quite far away from the expected correction; about with 500 cycles, the convergence becomes stable, and it doesn't need to do any training, at this point we arrive the best number of trainings. Figure 8(b) is the training situation of radial basis network. This method can achieve rapidly a convergence with expected value, and better avoid the errors than the back propagation neural network. And this study finds a more effective method which can avoid the errors in the trainings more effectively. That is general regression neural network shown in Fig. 8(c). This method doesn't have a smaller number of trainings but can correct more effectively the errors in the trainings. If we continue to do the trainings, the number of trainings will be excessive. When the


Universal COM debugging V1.0

Reception area

1D 1D 1D 1D 1D 1E 1E 1D

COM: COM1

BaudRate: 9600

Parity: None

Automatically send cycle: 1000 ms

Receipt: 0 Send: 0

Fig. 7. Interface of the collected temperature data.

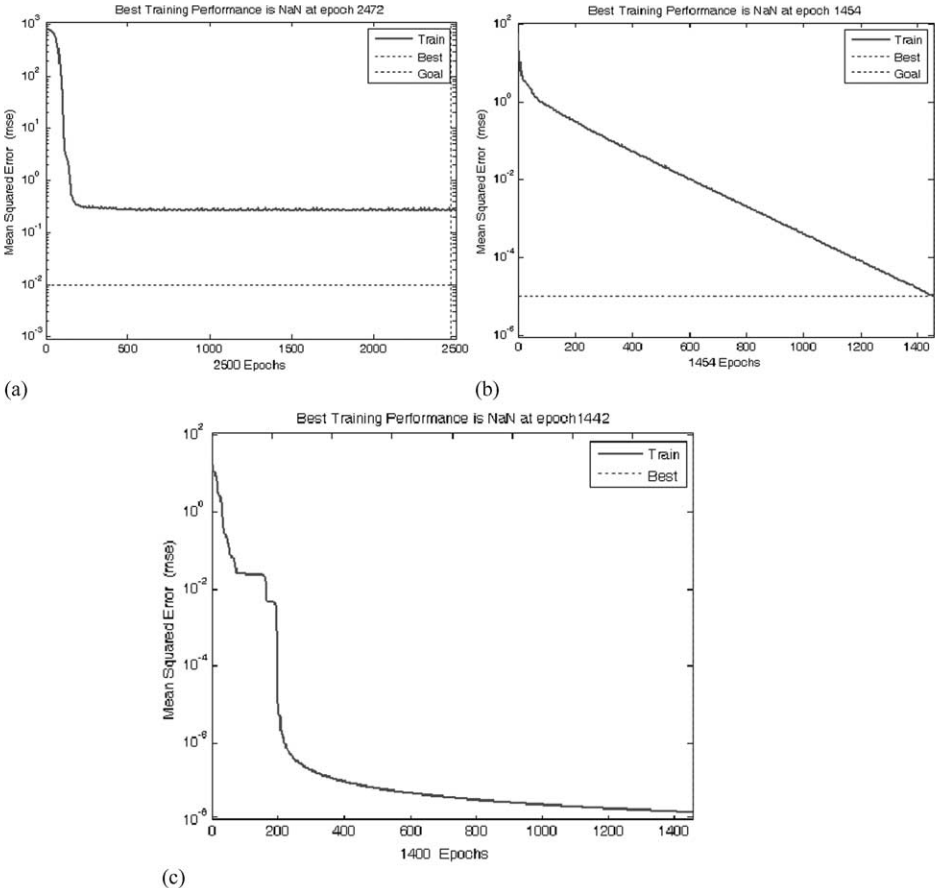


Fig. 8. (a) BP network training times, (b) RBF training times, (c) GRNN training times (remarks): y-axis for Mean Squared Error (MSE), x-axis one degree for 200 times-training data).

training data have reached the goal, we take the trained data to distinguish the accurate temperature data. This system can improve the system accuracy and reduce redundancy.

We can see the network input and output sample function carry out a preliminary and approximate simulation. Although it has some errors, it has obtained the approximate effect. If we estimate and find out the network simulation response curve, then compare it with radial basis function network, it shows that it's closer to the actual situation. In Fig. 9(a) BP linear regression results, we find that the output value and expected value do not coincide very well with the target, and the correlation coefficient is only 0.54. Then this study uses other methods, as shown in Fig. 9(b), RBF regression results expectation response and the simulation output regression effect are very good, and the correlation coefficient is 0.87, and, as shown in Fig. 9(c) about GRNN regression results, where the correlation coefficient is higher as 0.896. That's why this study uses GRNN to find out effectively all target values. In the operations, GRNN adaptability change is shown as Fig. 10, and the best individual $\sigma = 0.00243$.

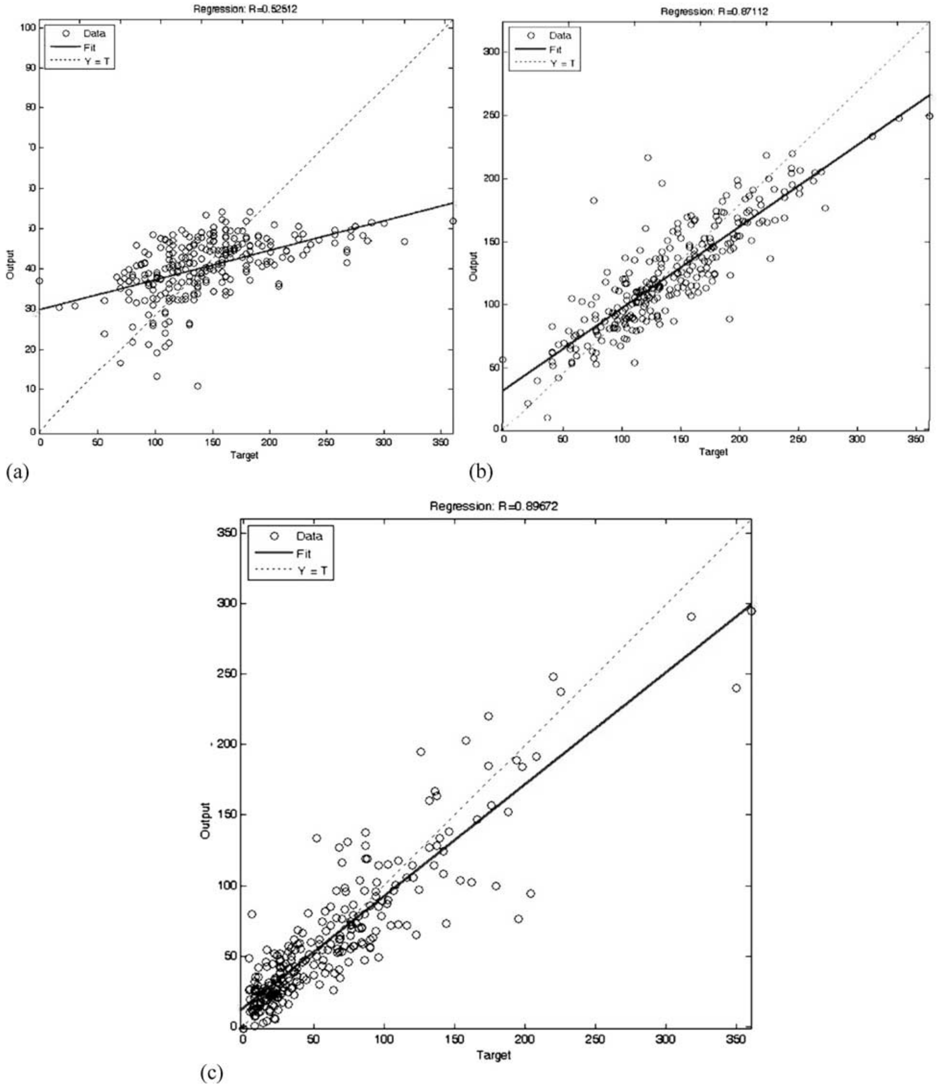


Fig. 9. (a) BP regression results, (b) RBF regression results, (c) GRNN regression results (remarks): y-axis for the actual data training output, x-axis for the target expected output).

This study illustrates the possible performance gains with the derived optimal multi-sensor precise measurement for the data fusion allocation scheme.

In the wireless sensors system experimental analysis, this work assumes that n_s is equal to 20, 50, 80, and 120 as the number of multiple wireless sensors. According to the experiment result, with a larger number of sensor signals, the proposed IHPG algorithm allows for more accurate measurement results (compared with the normal mode and the conventional PSO, especially when the experimental results are more obvious after $n_s > 75$.

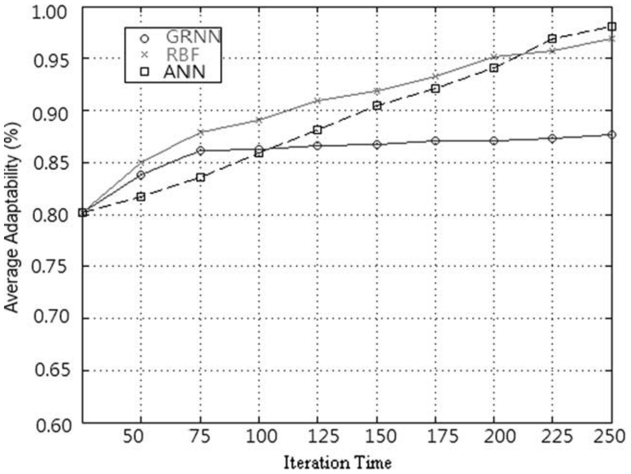


Fig. 10. Adaptability optimization of general neural network via IHPG based on smoothing parameter.

Table 1

Comparative analysis of calculated various initial populations and iterations in wireless sensors network experimental ($n_s = 20,\ 50,\ 80,\ 120$, IHPG algorithm)

n_s	Approaches	Average error rate (%)	Convergence (%)	Iterations (%)	Success (%)
20	General-PSO	< 0.18	91.0	49.5	93.6
	PSO-GA	< 0.11	93.6	47.1	90.3
	IHPG	$\mathbf{< 0.01}$	**98.7**	**62.3**	**90.2**
50	General-PSO	< 0.24	83.3	58.1	86.2
	PSO-GA	< 0.25	88.1	58.2	83.5
	IHPG	$\mathbf{< 0.15}$	**97.1**	**69.1**	**91.3**
80	General-PSO	< 0.35	75.6	69.2	85.1
	PSO-GA	< 0.48	84.5	65.2	86.9
	IHPG	$\mathbf{< 0.32}$	**96.3**	**71.2**	**95.3**
120	General-PSO	< 0.75	67.8	82.6	80.2
	PSO-GA	< 0.56	79.3	76.1	85.7
	IHPG	$\mathbf{< 0.26}$	**90.1**	**75.3**	**94.7**

When this study employed the IHPG algorithm 60 to 80 iterations were used to obtain the first system improvement. The situation becomes more obvious from the final integration of environment signals that the illumination sensor concentration measurement status is a relatively small error condition. It is relatively stable because the illumination sensor measurement is the optical measurement project, which is a more precise and difficult to changed measurement.

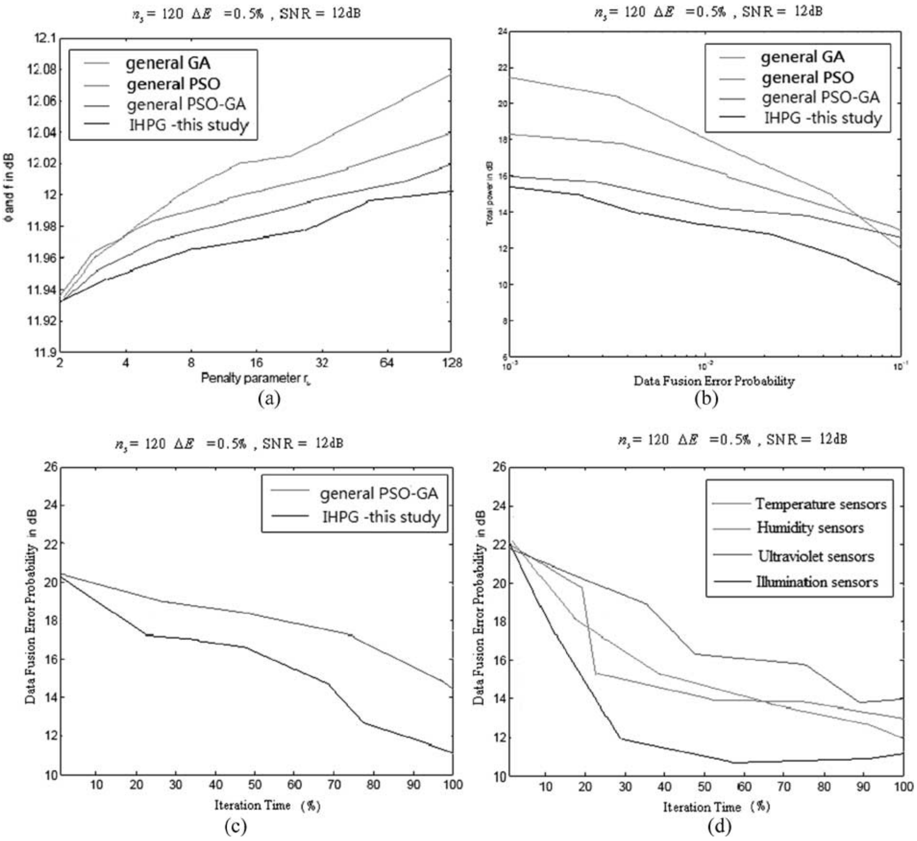


Fig. 11. Convergence of exterior penalty function based improved PSO: multi-sensor data fusion error probability is below 0.01%. (a) Optimum fitness returned for IHPG iterations for a given penalty parameter. (b) PSO data fusion error probability is improved when observations are correlated with each other. (c) Comparison the general PSO-GA with IHPG algorithm in data fusion error probability. (d) IHPG when various sensors behaviors in error probability (= 120; $\Delta E = 0.5\%$, SNR = 12dB).

8. Conclusion

IHPG is a hybrid evolutionary algorithm based on PSO and GA, which uses the GA search of randomness while maintaining the search performance of PSO. End of the paper by three standard neural network training function optimization via IHPG algorithm based on otimization of smoothing parameter, IHPG performance is much better than the general PSO-GA hybrid evolutionary algorithm. This study comparised the ANN, RBF, and GRNN three neural network in training time and regression analysis, GRNN 's network link weights adopts BP algorithm that the activity function of the nodes in hidden layers of GRNN adopts Gaussian function and thus GRNN has partial approximation ability and can learn quickly. GRNN has very few man-made adjustable parameters, it has only one threshold, and its network learning completely depends on the data samples. This feature allows GRNN to avoid the maximum effect on the predicted results caused by man-made subjective assumptions. This study uses ZigBee module, reduces greatly the cost of the wireless sensor module, and carries out the environmental monitoring of the region, sets up a large number of sensor nodes to carry out the transmission with respect to the coordinator. Lots of transmissions at the same time will easily produce lots of computations, thus this study uses neural network to carry out the calculation. And this paper also compares RBF Network with other neural networks. This study collects detected data and deals with the detected data and the weight distribution, then uses the neural network tool MATLAB to carry out the training and testing, and uses the previously built samples to do the training, and it's better than the original model. The RBF Network, which is applied to the feature level fusion, especially to the multi-dimensional feature level fusion, can be applied to the data processing before the fusion and suitable for hardware implementation.

Neural network training time by IHPG algorithm based on smoothing parameter optimization, with its generalization ability, high stability, good fault tolerance, and rapidity and effectiveness, can improve the system's engineering possibility and real-time capability. This study uses ANN, GRNN and RBF to compare with the currently commonly applied in IHPG, and it's obviously that for the estimated regional four type sensors (Temperature sensors, Humidity sensors, Ultraviolet sensors and Illumination sensors), GRNN can more wisely carry out the monitoring of the regional various changed. Although there are many algorithms that can be used for the feature-level data fusion, they have their own advantages and disadvantages, and these algorithms usually emphasize only one aspect. This study proposes the IHPG, and through computer simulation it is found that this algorithm can obtain a better fusion result than the simpler neural network because it enhances the fusion accuracy and provides data fusion technology with a computational intelligence algorithm.

Acknowledgements. The authors would like to thank the National Chin-Yi University of Technology, Taiwan for financially supporting this research. This research was supported by the National Science Council of Taiwan under grant NSC 99-2220-E-167-001-, NSC-100-2218-E-167-001- and NSC-100-2622-E-167-019-CC3.

References

Bhattacharyya S., Srikanthan T., Pramod, K. (2001). Ideal GMM parameters&posterior log likelihood for speaker verification. In: *Proceedings of IEEE Signal Processing Society Workshop*, 10–12, pp. 471–480.

Blum R., Kassam S., Poor H.V. (1997). Distributed detection with multiple sensors. Part II. Advanced topics. In: *Proceedings of the IEEE*, 85(1), pp. 64–79.

Bowden, G.J. *et al.* (2006). Forecasting chlorine residuals in a water distribution system using a general regression neural network. *Mathematical and Computer Modeling*, 44, 469–484.

Chair, Z., Varshney, P.K. (1986). Optimal data fusion in multiple sensor detection systems. In: *IEEE Transactions on Aerospace and Electronic Systems*, AES-22, pp. 98–101.

Chiuderi, A., Stafano, F., Cappellini, V. (1994). An application of data fusion to landcover classification of remote sensed imagery: a neural network approach. In: *Proceedings of IEEE International Conference Multisensor Fusion Integration Intelligent Systems*, pp. 756–762.

Dayhoff, J. (1990). *Neural Networks Architecture*. Van Nostrand Reinhold, New York.

Deng Wu, Chen Rong, Gao Jian, Song Yingjie, Xu Junjie (2012). A novel parallel hybrid intelligence optimization algorithm for a function approximation problem. *Computers&Mathematics with Applications*, 63(1), 325–336.

Fausset, L. (1994). *Fundamental of Neural Networks*. Prentiace Hall, New York.

Gandelli, A., Grimaccia, F., Mussetta, M., Pirinoli, P., Zich, R. (2007). Development and validation of different hybridization strategies between GA and PSO. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2782–2787.

Jagyasi, B.G., Dey, B.K., Merchant, S.N., Desai, U.B. (2006). Weighted aggregation scheme with lifetime-accuracy tradeoff in wireless sensor network. In: *Proceedings of 4th International Conference on Intelligent Sensing and Information Processing*.

Kao, T.-T., Zahara, E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing*, 8(2), 1942–1948.

Kuo, R.J., Han, Y.S. (2011). A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem – a case study on supply chain mode. *Applied Mathematical Modeling*, 35(8), 3905–3917.

Matthew, S., Terence, S. (2005). Breeding swarms: a GA/PSO hybrid. Genetic and evolutionary computation. In: *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, Washington, pp. 161–168.

Pohl, C., van Genderen, J.L. (1998). Review article. Multisensor image fusion in remote sensing: concepts, methods and applications. *International Journal of Remote Sensing*, 19(5), 823–854.

Seng, T.L., Khalid, M., Yusof, R. (2002). Adaptive GRNN for the modeling of dynamic plants. In: *Proceedings of the IEEE International Symposium on Intelligent Control*, Vol. 11, pp. 27–30.

Shahram, A., Napsiah, I. (2009). An improved algorithm for layout design in cellular manufacturing system. *Journal of Manufacturing Systems*, 28(4), 132–139.

Shahram, J., Gholam, S. (2012). PSO-SFDD: Defense against SYN flooding DoS attacks by employing PSO algorithm. *Computers&Mathematics with Applications*, 63(1), 214–221.

Shi, X.H., Lu, Y.H, Zhou, C.G., Lee, H.P., Lin, W.Z., Liang, Y.C. (2003). Hybrid evolutionary algorithm based on PSO and GA. In: *The 2003 Congress on Evolutionary Computation*, Vol. 4, pp. 2393–2399.

Specht, D.F. (1991). A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6), 568–576.

Srinivas, M, Patnaik, L.M. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 24(4), 656–667.

Sung, W.-T. (2010). Multi-sensors data fusion system for wireless sensors networks of factory monitoring via BPN technology. *Expert Systems with Applications*, 37(3), 2124–2131.

Sung, W.-T., Liu, Y.-F., Chen, J.-H., Chen, C.-H. (2010a). Enhance the efficient of WSN data fusion by neural networks training process. In: *International Symposium on Computer, Communication, Control and Automation*, Vol. 2, pp. 373–376.

Sung, W.-T., Liu, Y.-F., Chen, J.-H., Chen, C.-H. (2010b). *Enhance the Efficient of WSN Data Fusion by Neural Networks Training Process*. Department of Electrical Engineering, National Chin-Yi University of Technology, Taiwan, 373–376.

Sung, W.-T., Hsu, Y.-C. (2011a). Designing an industrial real-time measurement and monitoring system based on embedded system and ZigBee. *Expert Systems with Applications*, 38(4), 4522–4529.

Sung, W.-T., Tsai, M.-H. (2011b). Multi-sensor wireless signal aggregation for environmental monitoring system via multi-bit data fusion. *Applied Mathematics&Information Sciences*, 5(3), 589–603.

Valdez, F., Melin, P., Castillo, O. (2010). An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms. *Applied Soft Computing*, 11(2), 2625–2632.

Viswanathan, R., Varshney, P.K. (1997). Distributed detection with multiple sensors. Part I – fundamentals. In: *Proceedings of the IEEE*, Vol. 85(1), pp. 54–63.

Welstead, S.T. (1994). *Neural Network and Fuzzy Logic Applications in C/C++*. Wiley, New York.

Wu, Z.X., Chen, S.Q., Wang, J.W., Wang, L.F. (2012). Analysis of FBG reflection spectrum with PSO algorithm. *Applied Mechanics and Materials*, 182–183, 1953–1957.

Xing, C., Li, Y., Zhang, K., Wang, L. (2011). Shadow detecting using particle swarm optimization and the Kolmogorov test. *Computers&Mathematics with Applications*, 62(7), 2704–2711.

Yong Fang (2000). The research of data fusion from multiple sources by evidential reasoning. *Journal of Remote Sensing*, 4(2), 106–111.

Yu, L., Wang, N., Meng, X. (2005). Real-time forest fire detection with wireless sensor networks. In: *IEEE Proceedings of WCNM2005*, Wuhan, China, pp. 1214–1217.

Zhang, P., Ding, S.X., Frank, P.M., Sader, M. (2004). Fault detection of networked control systems with missing measurements. In: *Control Conference 5th Asian*, pp. 1258–1263.

Zurada, L. (1992). *Introduction to Artificial Neural Systems*. West Publisher, USA.

W.-T. Sung is working with the Department of Electrical Engineering, National Chin-Yi University of Technology as an associate professor. He received a PhD and MS degree from the Department of Electrical Engineering, National Central University, Taiwan in 2007 and 2000. He has won the 2009 JMBE Best Annual Excellent Paper Award and the dragon thesis award that sponsor is Acer Foundation. His research interests include wireless sensors network, data fusion, system biology, system on chip, computer-aided design for learning, bioinformatics, and biomedical engineering. He has published a number of international journal and conferences article related to these areas. Currently, he is the chief of Wireless Sensors Networks Laboratory.

C.-L. Hsiao is a master student at the Department of Electrical Engineering, National Chin-Yi University of Technology. His current research activities include the wireless sensors networks, SoC system and data fusion, etc. Currently, he is the member of Wireless Sensors Networks Laboratory.

Efektyvios informacijos sintezės GHKG algoritmas daugiasensoriniame tinkle, grįstas glotnia parametrų optimizacija

Wen-Tsai SUNG, Ching-Li HSIAO

Tyrime siūlomas inovatyvus pagerintas hibridinis algoritmas (GHKG) apjungiantis populiacijų optimizavimo ir genetinį algoritmus. Tyrimas apima kokybinį monitoringo eksperimentą panaudojant tris žinomus neurotinklų metodus bevielių sensorinių modulių duomenų apdorojime. Eksperimentas parodė kaip galima pagerinti įvertinamų duomenų tikslumą ir sumažinti skaičiavimų atsitiktinumą glotnių parametrų derinimo optimizacijos dėka.