Available online at www.sciencedirect.com

SciVerse ScienceDirect

AASRI Procedia 1 (2012) 183 – 188

AASRI Procedia

www.elsevier.com/locate/procedia

2012 AASRI Conference on Computational Intelligence and Bioinformatics

A HAMPSO-RBF Algorithm Applied to Target Localization

Jinjie Yao, Jing Yang, Liming Wang, Yan Han[*]， Jinxiao Pan

National Key Laboratory of Electronic Testing Technology, North University of China, Taiyuan, Shanxi, China

Abstract

This paper proposed a radial basis function neural network optimization algorithm with a hybrid adaptive mutation particle swarm. During the optimization of RBF neural networks, the HAMPSO method is adopted to train the network structure and applied to solve the problems of the target localization. The HAMPSO algorithm is a dynamically adaptive optimization approach using uniform distribution mutation and Gaussian distribution mutation to escape local optima. We propose a HAMPSO method that can expedite convergence toward the global optimum during the iterations. In order to verify that the proposed HAMPSO-RBF approach has effect, comparisons with the RBF, genetic algorithm based RBF and PSO-based RBF approach are made. The computational results proved that the proposed HAMPSO-RBF approach exhibits much better and faster convergence performance in the training process as well as better prediction ability in the validation process than the results of other three approaches.

Keywords: Radial basis function networks; Target localization; Time difference of arrival; Particle swarm optimization

1. Introduction

Target localization based on time difference of arrival is an vital issue in many applications such as radar, sonar and wireless sensor network [1]. In the TDOA localization method, the differences in arrival times of the source signal at multiple pairs of base stations are measured, the target position is achieved by calculating the distance difference between the target and other different reference base stations. When base stations are distributed arbitrarily, the position of target is determined from the intersection of a set of hyperbolic curves by the TDOA estimates, and the problem is hard to solve due to the noise of the environment and the high nonlinearity of localization equations.

doi:10.1016/j.aasri.2012.06.029

The literature demonstrated that the target position can be effectively obtained using RBFNNs [2]. However, optimizing the structure of RBFNNs is still a challenging work. This paper proposed a RBFNNs optimization algorithm by HAMPSO and applied to target localization. The feasibility and efficiency of the algorithm over the PSO-RBFNNs and the GA-RBFNNs is demonstrated in the computational results.

Nomenclature

RBFNNs	radial basis function neural networks
HAMPSO	hybrid adaptive mutation particle swarm optimization
GA-RBF	genetic algorithm based RBF
PSO	particle swarm optimization
PSO-RBF	PSO-based RBF approach
TDOA	time difference of arrival
ANN	artificial neural network
RBFNN	radial basis function neural network
PSO-RBFNNs	RBFNNs based on PSO
GA-RBFNNs	RBFNNs based on Genetic Algorithm

2. Principles

In this part, it will focus on introduce the basic principle of target localization based on TDOA and what is the RBFNNs. This paper will be based on the principle to put forward a improved algorithm.

2.1. Target localization method

The base stations are distributed in 3-D space by stated rule, as shown in Fig. 1.

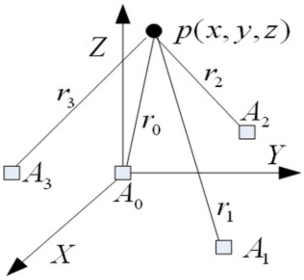

Fig. 1. Target localization sketch map

In the Fig.1, the target position *is denoted as* p(x,y,z), the position of *i*th base station *is denoted as* $A_i(x_i,y_i,z_i)$, and the mathematic model of target localization can be expressed as (1). Where r_i is the distance between the target and base station A_i, $r_{i,0}$ is the distance difference from the base station A_i between the target and base station A_i (i=1,2,3). c is the transmission speed, $\tau_{i,0}$ is the time difference value, $n_{i,0}$ is the measurement noise. The target position is the calculation result of the equations [3].

$$\begin{cases} r_0^2=(x-x_0)^2+(y-y_0)^2+(z-z_0)^2 \\ r_i^2=(x-x_i)^2+(y-y_i)^2+(z-z_i)^2 \\ r_{i,0}=r_i-r_0+cn_i=c\tau_{i,0} \\ i=1,2,3 \end{cases} \tag{1}$$

2.2. RBFNNs

In RBFNNs, the Gaussian function is used as radial basis function with the following form,

$$\phi_j(x)=\exp\left(-\frac{(x-c_j)^2}{2\sigma_j^2}\right) \tag{2}$$

where $\phi_j(x)$ is the *j*th radial basis function, $x=(x_1,x_2,\cdots,x_d)^T$ is the input vector, c_j and σ_j^2 are the *j*th center vector and the width parameter, respectively. The output of RBF network y which is the linear sum of radial basis function is given as follows [4]:

$$y=\sum_{j=1}^{p} w_j\phi_j(x) \tag{3}$$

where y is the output of the RBF neural network, p is the number of the hidden layer neuron, and w_j is the weight from *j*th neuron to the output layer.

3. PSO and its improvement

PSO algorithm and the principle of improved PSO algorithm, namely the HAMPSO algorithm, will be expounded in this part in detail.

3.1. PSO Algorithm

In PSO, let the position and velocity of the *i*th particle be expressed by $X_i=[x_{i1}, x_{i2}, \cdots, x_{in}]$ and $V_i=[v_{i1}, v_{i2}, \cdots, v_{in}]$ in the *D* dimensional search space [5]. Meanwhile, according to the actual optimization problem, let the local optima of the *i*th particle be denoted as $P_i=[p_{i1}, p_{i2}, \cdots, p_{in}]$ and the global best position found so far be denoted as $P_g=[p_{g1}, p_{g2}, \cdots, p_{gn}]$. At each step, the velocity and position updating rule is given by:

$$V_i=\omega\cdot V_i+c_1r_1(P_i-X_i)+c_2r_2(P_g-X_i) \tag{4}$$

$$X_{i+1}=X_i+V_i \tag{5}$$

where c_1 and c_2 are constants named acceleration coefficients. r_1 and r_2 are two independent random numbers uniformly distributed in the range of $(0,1)$. ω is called the inertia weight that controls the impact of previous velocity of particle on its current one [6].

3.2. HAMPSO Algorithm

In the HAMPSO algorithm, there is a hybrid mutation operator designed, which combines uniform distribution mutation and Gaussian distribution mutation [7]. The former prompts a global search in a large range while the latter searches in a small range with high precision. The hybrid mutation operator can be formally defined as

$$\begin{cases} x_{id} = x_{\min} + rand \times (x_{\max} - x_{\min}) & rand \le p_m \\ x_{id} = x_{id} \times (1 + 0.5 * Gaussian(\sigma)) & rand > p_m \end{cases} \tag{6}$$

In the formula, the ratio of the uniform distribution mutation is denoted as p_m, which can be self-adaptive. Then the ratio of the Gaussian distribution mutation is 1-p_m. The $Gaussian(\sigma)$ returns a random number drawn from a Gaussian distribution with a standard deviation of σ, and x_{min}, x_{max} are the minimum and the maximum. The proposed adaptive mutation operator is calculated as follows:

$$p_m = \begin{cases} p_{\max} - \dfrac{(p_{\max} - p_{\min})(f_{\max} - f)}{f_{\max} - f_{avg}}, f \ge f_{avg} \\ p_{\max}, f < f_{avg} \end{cases} \tag{7}$$

where f_{max} and f_{avg} are the maximum and the mean fitness of the particle swarms, f is the local best fitness of the particle, p_{max} and p_{min} are the maximum and the minimum component of the mutation probability, respectively.

4. Target Localization Based on HAMPSO-RBFNNs

To realize target localization based on HAMPSO-RBF technology, the process of target localization by HAMPSO-RBF should be divided into the following three steps.

4.1. Preparing experimental data

Set the position of stations is (0, 0, 0) m, (-12990, 7500, 200) m, (0, 15000, 150) m, (12900, 7500, 100) m. The flight time of the target is 140 seconds and the interval of the signal transmission is one second. The locations of aerial target are also obtained by using GPS. There are 120 sets of time-differences and target positions collected from the localization measure system, among which 100 sets of data are used for RBF neural network's training while the rest are for verification of the position prediction. Because the obtained time differences are very tiny and the coordinates of the target are large comparatively, all the data, including time differences and the coordinates, are scaled to [0, 1] by Eq. (10).

$$\begin{cases} new_{TDOA} = old_{TDOA} \times 10^{5} \\ new_{coordinate} = old_{coordinate} \times 10^{-5} \end{cases} \tag{8}$$

where new_{TDOA} is the scaled time difference, old_{TDOA} is the measured time difference, $new_{coordinate}$ and $old_{coordinate}$ are the scaled and measured coordinates, respectively.

4.2. Optimization operation

In order to reduce parameter number and improve the speed of program debug, the number of neuron in the hidden layer of the RBF neural network is 20. There are three inputs and three outputs, so the structure of the RBF neural network is 3-20-3.

The HAMPSO algorithm that trains RBFNNs is described as follows:

Step 1. PSO parameters setting: the population size is 20, the inertia weights $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.2$, the number of generations is 100, the initial velocity range of particles in PSO is [-1, +1]. The mutation probability $p_{max} = 0.1$ and $p_{min}= 0.01$. The acceleration coefficients c_1 and c_2 are both 2.0.

Step 2. Set iterative variable: $k = 0, k = k + 1$.

Step 3. Compute the fitness value of each particle, which is defined as follows:

$$Fitness = SSE = \sum_{i=1}^{n} \left(Av_i - Ev_i \right)^2 \tag{9}$$

where Ev_i is the estimated value and Av_i is the actual value. Take current particle as individual extreme point of every particle and do the particle with minimal fitness value as the global extreme point.

Step 4. Stop condition checking: if maximum iterations are met, go to Step 7. Otherwise, go to the next step.

Step 5. Update the particle position and velocity by Eq. (4) and (5), and then form new particle swarms.

Step 6. Adopt the adaptive mutation operator by Eq. (7), and manipulate the particle position according to Eq. (6). Go to *Step* 2.

Step 7. The training over, and output the optimal particles.

4.3. Result analysis

After training, a HAMPSO-RBF model is obtained, which can be used to predict new input data. A comparison among GA-RBF, PSO-RBF and HAMPSO-RBF is made to evaluate the convergence performance, which is shown in Fig.2.

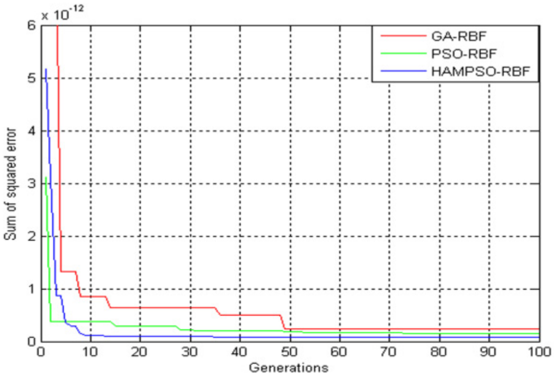

Fig.2 comparing the error during training process

It can be observed from Fig.2 that the HAMPSO-RBF algorithm performs better than the GA-RBF algorithm and PSO-RBF algorithm in terms of convergence speed. The root mean squared error (RMSE) of the testing data for target localization are shown in Fig. 3 through the RBF, GA-RBF, PSO-RBF and HAMPSO-RBF algorithm. As shown in Fig. 3, the maximal RMSE by adopting the HAMPSO-RBF algorithm is less than 2.5m in the search region for which the diagonal line length is about 15km. However, the maximal RMSE by using the RBF algorithm has reached 10m, and it reaches 3.65m and 3.45m by GA-RBF and PSO-RBF algorithm, respectively. It has been shown that the HAMPSO-RBF's RMSE performance is superior to the responses of RBF, GA-RBF and PSO-RBF algorithm.

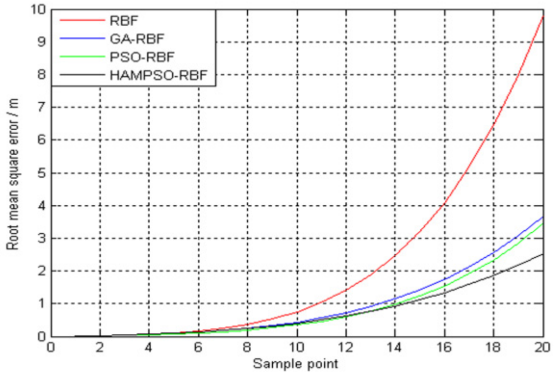

Fig.3 the roots mean squared error analysis

5. Conclusions

A prediction of the RBF neural network for target localization was presented by using the HAMPSO method. In the proposed method, the HAMPSO technique is used to optimize the RBF neural network parameters. The computational results indicate that the proposed method is more efficient than RBF, GA-RBF and PSO-RBF methods. The proposed HAMPSO-RBF method can be further extended to the application in other scientific fields.

Acknowledgements

This paper is supported by National Key Laboratory of Electronic Testing Technology Foundation of China under Grant No.9140C1204051010 and School youth science fund.

References

[1]Kenneth W.K. Lui, H.C. So, A study of two-dimensional sensor placement using time difference of arrival measurements, Digital Signal Processing, 19 (2009) 650-659.
[2]W.Q. Guo, T.S. Qiu, H. Tang, W.R. Zhang, Performance of RBF neural networks for array processing in impulsive noise environment, Digital Signal Processing, 18 (2008) 168-178.
[3]Adrian N. Bishop, Baris Fidan, Kutluyil Dogancay, Brian D.O. Anderson, Pubudu N. Pathirana, Exploiting geometry for improved hybrid AOA/TOA-based localization, Signal Processing, 88 (2008) 1775-1791.
[4]N. Qu, H. Mi, B. Wang, Y.L. Ren, Application of GA-RBF networks to the nondestructive determination of active component in pharmaceutical powder by NIR spectroscopy, Journal of the taiwan Institute of Chemical Engineers, 40 (2009) 162-167.
[5]V.M. Rivas, J.J. Merelo, P.A. Castillo, M.G. Arenas, J.G. Castellano, Evolving RBF neural networks for time-series forecasting with EvRBF, Information Sciences, 165 (2004) 207-220.
[6]Y. Liu, Z. Qin, Z.W. Shi, J. Lu, Center particle swarm optimization, Neurocomputing, 70 (2007) 672-679.
[7]X.W. Zheng, H. Liu, A hybrid vertical mutation and self-adaptation based MOPSO, Computers and Mathematics with Applications, 57 (2009) 2030-2038.