Neural Network for WGDOP Approximation and Mobile Location

Chien-Sheng Chen, Jium-Ming Lin, and Chin-Tan Lee

1 Department of Information Management, Tainan University of Technology, Tainan 71002, Taiwan
2 Department of Communication Engineering, Chung-Hua University, Hsinchu 30012, Taiwan
3 Department of Electronic Engineering, National Quemoy University, Quemoy 89250, Taiwan

Correspondence should be addressed to Chien-Sheng Chen; t00243@mail.tut.edu.tw

Received 12 April 2013; Revised 17 June 2013; Accepted 17 June 2013

1. Introduction

Mobile positioning is becoming an increasingly important problem, whose solutions can be generally divided into two major categories—handset-based methods and network-based methods [1]. When equipped with a global positioning system (GPS) receiver, handset-based location schemes require modifications to the handset to calculate its own position. The network-based location schemes can be used in situations where GPS signals are not available, for example, indoor environment, or when GPS-embedded handsets are not available. The network-based methods require the estimation of the mobile location based on the received signals between the mobile station (MS) and a set of base stations (BSs). For many applications in wireless sensor networks (WSN), such as environmental sensing and activities measuring, it is crucial to know the locations of the sensor nodes; this is known as a “localization problem” [1].

Geometric dilution of precision (GDOP) can be applied as a criterion for choosing the appropriate geometric configuration of the measurement units. Different stations can constitute different combinations, and the stations randomly selected can yield with relatively poor accuracy. Reference [2] proposed a method based on fuzzy clustering to analyze the positioning distribution of each combination. The key of this method is to apply only those stations with good GDOP and time of arrival (TOA). In reference [3] a time-varying function based on the GDOP curved surface is defined, so it is quite complex and difficult to effectively track a mobile target and maintain good GDOP. However, the proposed target localization scheme is to use the best mobile sensor nodes (MSNs) location, which can effectively mitigate the effect of GDOP and provide more accurate location estimates in mobile sensing systems. On the other hand, reference [4] also proposed a difficult hybrid algorithm, by combining pseudorange differenting and ridge regression technique, to improve both the position accuracy of pseudolite only system.
and effectively reduce the GDOP effects. But the proposed method in this paper can be applied for those conditions, such as in indoor navigation system design and when GPS satellite signals are not available.

This paper considers both network-based and handset-based methods, employing the concept of GDOP, which was developed to select the optimal geometric configuration of satellites in GPS. Thus when enough measurements are available, the optimal measurements selected can reduce the adverse geometry effects, thereby improving the location accuracy. However, excessive or redundant measurements would increase the computational overhead, and sometimes it may not improve the location accuracy significantly. So before positioning, it is very important to select a subset with the most reasonable measurement units rapidly.

Implementation of the GDOP method assumes that all pseudorange errors are independent and identically distributed [5]. But the measurements errors usually have different variances in practice [6]. Generally, the satellite signal is obtained by combining the values such as user range accuracy, carrier-to-noise ratio, elevation angle, and the ephemeris. In [7], Sairo et al. proposed a method that takes into account the different error variances. In [8], the elevation and receiver’s signal-to-noise ratio (SNR) is used to weigh GDOP and provide the positioning solution. When baro-altitude measurements or a priori terrain elevation information is used, the conventional GDOP formula cannot be applied and must be modified [9]. The combinations of GPS and Galileo satellite constellations will provide more visible satellites with better geometric distribution and their accessibility will be significantly improved. A weighted GDOP (WGDOP) algorithm was proposed in [10] for the combined GPS-Galileo navigation receiver. Reference [11] considered the value of the WGDOP, the number of visible satellites, and the constellation costs as three objective functions of navigation constellation performance. Simulation results show that these optimal solutions can provide more visible satellites and better WGDOP. In addition, several methods based on WGDOP have been proposed to improve the GPS positioning accuracy [12–18]. Most, if not all, of those methods need matrix inversion to calculate WGDOP. The matrix inversion method is rather time consuming and causes a great deal of computational burden. Thus those performances are achieved at the expense of increased computational complexity, which is usually too much to be practical.

Simon and El-Sherief [19, 20] employed backpropagation neural network (BPNN), a supervised learning neural network [21], to obtain an approximation to the GDOP function. The BPNN was employed to “learn” the relationship between the entries of a measurement matrix and the eigenvalues of its inverse. Three other input-output relationships were proposed and compared based on simulation results [22]. However, BPNN generally converges slowly and tends to get trapped in local minima. Considering both effectiveness and efficiency, this paper presents two novel architectures based on an alternative artificial neural networks method, namely, the resilient backpropagation (Rprop) method [21], to approximate WGDOP. Rprop is an algorithm with good convergence speed, accuracy, and robustness to the training parameter [23]. Comparing to the BPNN, the Rprop converges faster and needs less training. Fast convergence can curtail the training time, and build relevant predictive models of neural network quickly. But collecting the training data spends a lot of cost, time, and resources. A training pattern consists of a set of the input vectors and the corresponding output vectors. In many situations, the data for training is not available. The accuracy of prediction may be affected by the lack of sufficient training data. However, the more training data we have the more training time and resource it costs. It is very critical to achieve higher accuracy when the training data is not sufficient. Both faster convergence speed and less number of training iterations for neural network are very important. Simulation results have shown that the proposed architectures based on Rprop can provide faster convergence and less number of training iterations. So they can be applied to select the location measurement units based on Rprop in GPS, WSN, and cellular communication systems. In practice, the measurement units of GPS, WSN, and cellular communication systems are satellites, sensors, and BSs, respectively.

To select the most appropriate set of BSs to achieve the minimum positioning error, we need to consider not only GDOP effect but also the non-line-of-sight (NLOS) error statistic to each BS in cellular communication systems. The reciprocal of square root of an upper bound of NLOS errors is set as the weight coefficient. In this paper, we select a subset of four measurements in the location process. We will first select the serving BS as one of the four, and then combine it with three other measurements to form the WGDOP subsets. After calculating the WGDOP values, the subset with minimum WGDOP is used to estimate the MS location. Simulation results show that Rprop provides much better estimation of average WGDOP residual than those based on BPNN. The number of epochs of Rprop is less than the traditional BPNN. Rprop algorithm converges faster than BPNN and requires much less convergence time. The proposed architectures using Rprop for WGDOP approximation always yield better location results compared with the other architectures. The proposed architectures using Rprop for both WGDOP approximation and the matrix inversion method provide nearly identical MS location estimation. The four randomly selected BSs with poor WGDOP will give bad location estimation, and the positioning accuracy would be seriously affected by the geometric configuration of BSs and MS. The proposed BSs selection criterion reduces greatly the number of subsets while providing a comparable level of accuracy in location estimation. Therefore, the conclusion is that the proposed algorithm can be applied in practical situations.

To improve positioning accuracy even further, the higher precision MS locations according to the first several minimum WGDOP subsets can be used as the inputs to the next Rprop. After training, Rprop can be applied to predict the final MS location by means of the higher precision MS locations. The simulation results confirm that the proposed methods employing at most two higher MS locations can always perform better than using all seven BSs.
In essence, these methods greatly reduce the NLOS error and enhance the performance of MS location estimation effectively.

The remainder of this paper is organized as follows: Section 2 presents the calculation of GDOP and WGDOP. Section 3 describes briefly BPNN and Rprop methods. The six types of mapping for WGDOP approximation based on Rprop are proposed in Section 4. Section 5 presents a proposed BSs selection criterion and the location methods. Simulation results are given in Section 6 followed by conclusion in Section 7.

2. Calculation of GDOP and WGDOP

The concept of GDOP is commonly used to determine the geometric effect of GPS satellite configurations. It has a simple form if all the measurements have the same error variance. The smaller the GDOP value, the more accurate the positioning is. In order to improve the positioning accuracy, we should minimize GDOP among the selected measurement units.

Using a three-dimensional (3D) Cartesian coordinate system, the distances between satellite \(i\) and the user can be expressed as

\[
r_i = \sqrt{(x-X_i)^2 + (y-Y_i)^2 + (z-Z_i)^2 + C \cdot t_b + \nu_{ri}}, \tag{1}\]

where \((x, y, z)\) and \((X_i, Y_i, Z_i)\) are the locations of the user and satellite \(i\), respectively; \(C\) is the speed of light, \(t_b\) denotes the time offset, and \(\nu_{ri}\) is pseudorange measurements noise. Equation (1) can be linearized with Taylor’s series expansion at the approximate user position \((\tilde{x}, \tilde{y}, \tilde{z})\), neglecting the higher order terms. Defining \(\tilde{r}_i\) as \(r_i\) at \((\tilde{x}, \tilde{y}, \tilde{z})\), we can obtain

\[
\Delta r = r_i - \tilde{r}_i \equiv e_{i1}\delta_x + e_{i2}\delta_y + e_{i3}\delta_z + C \cdot t_b + \nu_{ri}, \tag{2}\]

where \(\delta_x, \delta_y, \delta_z\) are, respectively, the coordinate offsets of \(x, y, z\),

\[
e_{i1} = \frac{\tilde{x} - X_i}{\tilde{r}_i}, \quad e_{i2} = \frac{\tilde{y} - Y_i}{\tilde{r}_i}, \quad e_{i3} = \frac{\tilde{z} - Z_i}{\tilde{r}_i},
\]

\[
\tilde{r}_i = \sqrt{(\tilde{x} - X_i)^2 + (\tilde{y} - Y_i)^2 + (\tilde{z} - Z_i)^2}\]

\[
\left( e_{i1}, e_{i2}, e_{i3} \right), \quad i = 1, 2, \ldots, n. \tag{3}\]

The linearized equations can be expressed in a vector form with direction cosines from the user to the \(i\)th satellite

\[
z = H\delta + \nu, \tag{4}\]

where

\[
v = \begin{bmatrix} v_{r1} \\ v_{r2} \\ \vdots \\ v_{rn} \end{bmatrix}, \quad H = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ \vdots & \vdots & \vdots \\ e_{n1} & e_{n2} & e_{n3} \end{bmatrix} \tag{6}\]

is the geometry matrix.

The vector variable \(\delta\) in (4) can be solved with the least-square (LS) algorithm, namely,

\[
\tilde{\delta} = (H^T H)^{-1} H^T z. \tag{7}\]

Assuming that the pseudorange errors are uncorrelated with equal variances \(\sigma^2\), the error covariance matrix can be expressed as

\[
E \left[ (\tilde{\delta} - \delta)(\tilde{\delta} - \delta)^T \right] = \sigma^2 \cdot (H^T H)^{-1}. \tag{8}\]

The variances are functions of the diagonal elements of \((H^T H)^{-1}\). The GDOP is a measure of accuracy for positioning systems and is defined as

\[
\text{GDOP} = \sqrt{\text{tr}(H^T H)^{-1}}. \tag{9}\]

In practice, the measurement errors do not have the same variance, especially when different systems are combined. The covariance matrix has the form

\[
E \left( \nu \nu^T \right) = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_n^2 \end{bmatrix}. \tag{10}\]

Now, define a weight matrix \(W\) as follows,

\[
W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_1^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_2^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_n^2} \end{bmatrix} = \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & k_3 & 0 \\ 0 & 0 & 0 & k_n \end{bmatrix}. \tag{11}\]

where \(\sigma_i^2 = 1/k_i, i = 1, 2, \ldots, n\) are variances of measurement errors.

With the weighting matrix defined above, we now need to solve a weighted least-square (WLS) problem and the solution is given by

\[
\tilde{\delta} = (H^T W H)^{-1} H^T W z. \tag{12}\]

To select the set of the most appropriate measurement units that renders the minimum positioning error, we must consider not only the GDOP effect but also the ranging error statistics. In this paper, we employ WGDOP, instead of GDOP, to select measurement units so as to improve the
accuracy of location. The optimal subset is the one with the minimum WGDOP, which is given by the trace of the inverse of the $H^T WH$ matrix:

$$\text{WGDOP} = \sqrt{\text{tr}(H^T WH)^{-1}}. \quad (13)$$

The conventional method for calculating WGDOP is to use matrix inversion for all subsets, requiring a great deal of computational effort. When the number of measurement units increases, the computation time will increase rapidly. In this paper, we employ an artificial neural network learning algorithm to obtain approximate WGDOP.

3. Traditional BPNN Algorithm and Rprop Algorithm

It has been known that BPNN is capable of learning and realizing both linear and nonlinear functions [21]. The learning process of BPNN can be considered as one of gradient descent methods that minimizes some measure, for example, mean-square value of the difference between the actual output vector of network and the desired output vector. Define an error function $F$

$$F = \frac{1}{2} \sum_k (T_k - O_k)^2, \quad (14)$$

where $T_k$ is the output vector of the network while $O_k$ is the desired output vector. Then, the gradient decent algorithm is employed to adapt the weights (namely, synapses) as follows:

$$\Delta w_{ij} (t) = -\epsilon \frac{\partial F}{\partial w_{ij}} (t), \quad (15)$$

where $\epsilon$ is a predetermined learning rate, and $w_{ij}$ denotes the weight connecting neuron $i$ to neuron $j$. The major drawbacks of traditional BPNN include slow learning process and the tendency to be trapped easily in local minima.

Comparing to the traditional BPNN algorithm, the Rprop algorithm offers faster convergence and is usually more capable of escaping from local minima. In a sense, Rprop is a first-order algorithm and its time and memory requirement scales linearly with the number of parameters. In practice, Rprop is easier to implement than BPNN. Besides, a hardware implementation for Rprop has been presented in [24]. Briefly speaking, Rprop performs a direct adaptation of the weighting step based on local gradient information. The main idea of Rprop is to reduce the potential spurious effect of the partial derivative on weight-updates by retaining only the sign of the derivative as an indication of the direction, in which the error function will be changed by the weight-update. We introduce for each weight an individual update-value $\Delta_{ij}(t)$, which solely determines the size of the weight-update. This adaptive update-value evolves during the learning process based on its local sight on the error function $F$, according to the following learning rule [25]:

$$\Delta_{ij}(t) = \begin{cases} 
\eta^+ \cdot \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial F}{\partial w_{ij}} \cdot \frac{\partial F^{(t)}}{\partial w_{ij}} > 0 \\
\eta^- \cdot \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial F}{\partial w_{ij}} \cdot \frac{\partial F^{(t)}}{\partial w_{ij}} < 0 \\
\Delta_{ij}^{(t-1)}, & \text{else,}
\end{cases} \quad (16)$$

where $0 < \eta^- < 1 < \eta^+$. We can simply describe the adaptation rule as follows: every time the partial derivative of the error function $F$ with respect to the corresponding weight $w_{ij}$ changes its sign, the update-value $\Delta_{ij}$ is decreased by a factor $\eta^-$. If the derivative retains its sign, the update-value is slightly increased in order to accelerate convergence in shallow regions.

Once the update-value for each weight is adapted, the weight-update itself follows a very simple rule: if the derivative is positive (increasing error), the weight is decreased by its update-value, and if the derivative is negative, the update-value is added to the weight:

$$\Delta w_{ij}^{(t)} = \begin{cases} 
-\Delta_{ij}^{(t)}, & \text{if } \frac{\partial F^{(t)}}{\partial w_{ij}} > 0 \\
+\Delta_{ij}^{(t)}, & \text{if } \frac{\partial F^{(t)}}{\partial w_{ij}} < 0 \\
0, & \text{else.}
\end{cases} \quad (17)$$

4. Proposed Network Architectures for WGDOP Approximation

Researchers have employed the conventional BPNN to estimate GDOP; see, for example, [19, 20, 22]. This can reduce the computational complexity required to compute the matrix inversion in calculating GDOP. Since the statistics of different location measurement units are, in general, not equal to each other, WGDOP serves as an index for the precision of location in different networks, such as GPS, WSN, and cellular communication systems. In this paper, the original four types BPNN, defined by four different input-output mapping, for GDOP calculation [19, 20, 22] will be extended to WGDOP with the employment of Rprop. In addition, we propose two new mapping architectures.

To further reduce the computational overhead and improve location performance, the selection of optimal measurement units is necessary. Instead of using all visible satellites, four satellites are usually sufficient for GPS positioning. As such, we take only four BSs from among seven with better geometry to estimate the MS location in cellular communication networks. The different structures of four location measurement units were implemented to illustrate the applicability of Rprop for WGDOP predictions in 3D environments. Accordingly, from (13), $H^T WH$ is a $4 \times 4$ matrix and it has four eigenvalues, namely, $\lambda_i, i = 1, 2, 3, 4$. 
Therefore, the four eigenvalues of \((H^T WH)^{-1}\) are \(\lambda_i^{-1}, i = 1, 2, 3, 4\). Consequently, WGDOP can be expressed as

\[
WGDOP = \sqrt{\text{tr} (H^T WH)^{-1}} = \sqrt{\lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1} + \lambda_4^{-1}}. 
\] (18)

On the other hand, the geometry matrix and weight matrix composed of four location measurement units in two-dimensional (2D) environments are

\[
H = \begin{bmatrix}
e_{11} & e_{12} & 1 \\
e_{21} & e_{22} & 1 \\
e_{31} & e_{32} & 1 \\
e_{41} & e_{42} & 1
\end{bmatrix},
\]

\[
W = \begin{bmatrix}
k_1 & 0 & 0 & 0 \\
0 & k_2 & 0 & 0 \\
0 & 0 & k_3 & 0 \\
0 & 0 & 0 & k_4
\end{bmatrix},
\] (19)

respectively.

Therefore, \(H^T WH\) is a \(3 \times 3\) matrix and WGDOP is

\[
WGDOP = \sqrt{\text{tr} (H^T WH)^{-1}} = \sqrt{\lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1}}. 
\] (20)

In the following, we present six types of Rprop mapping architectures for WGDOP prediction in 3D and 2D environments and the mapping relationship with the three layer “input-p-hidden neuron number-output q” structures. These six types of architectures are described by a block diagram as shown in Figure 1.

**Type 1.** (A) 3D: four inputs are mapped to four outputs

\[
f_1 (\lambda) = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = \text{trace} (H^T WH), 
\] (21a)

\[
f_2 (\lambda) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 + \lambda_4^2 = \text{trace} \left( (H^T WH)^2 \right), 
\] (21b)

\[
f_3 (\lambda) = \lambda_1^3 + \lambda_2^3 + \lambda_3^3 + \lambda_4^3 = \text{trace} \left( (H^T WH)^3 \right), 
\] (21c)

\[
f_4 (\lambda) = \lambda_1 \lambda_2 \lambda_3 \lambda_4 = \det (H^T WH). 
\] (21d)

The network has the input-output pairs:

**Input:** \((f_1, f_2, f_3, f_4)^T\)

**Output:** \((\lambda_1^{-1}, \lambda_2^{-1}, \lambda_3^{-1}, \lambda_4^{-1})^T\).

One can see that the mapping from \(f\) to \(\lambda^{-1}\) is nonlinear, which is usually difficult to be determined analytically. After the training period, this mapping relationship can be approximated quite well by neural network. WGDOP is estimated by taking the square root of the sum of the outputs.

**Type 2.** (A) 2D: three inputs are mapped to one output

\[
f_1 (\lambda) = \lambda_1 + \lambda_2 + \lambda_3 = \text{trace} (H^T WH), 
\] (22a)

\[
f_2 (\lambda) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 = \text{trace} \left( (H^T WH)^2 \right), 
\] (22b)

\[
f_3 (\lambda) = \lambda_1 \lambda_2 \lambda_3 = \det (H^T WH). 
\] (22c)

The network has the following input-output pair:

**Input:** \((f_1, f_2, f_3)^T\)

**Output:** \((\lambda_1^{-1}, \lambda_2^{-1}, \lambda_3^{-1})^T\).

The sum of the outputs gives the square of WGDOP.

**Type 3.** (A) 3D: ten inputs are mapped to four outputs. In this case, WGDOP values are directly used as the output of the training data. The network has the following structure:

**Input:** \((f_1, f_2, f_3, f_4)^T\)

**Output:** WGDOP.

**Type 4.** (A) 2D: three inputs are mapped to four outputs.

**Input:** \((f_1, f_2, f_3)^T\)

**Output:** WGDOP.

**Type 5.** (A) 3D: ten inputs are mapped to four outputs. This type of mapping is used to obtain approximations to the inverse of the eigenvalues from the elements of the matrix \((H^T WH)\). The matrix \((H^T WH)\) is a symmetric matrix and can be expressed as

\[
H^T WH = \begin{bmatrix}
B_{11} & B_{12} & B_{13} & B_{14} \\
B_{22} & B_{23} & B_{24} \\
B_{33} & B_{34} \\
B_{44}
\end{bmatrix},
\] (23)

where \(B_{ij} = k_1 e_{1j}^2 + k_2 e_{2j}^2 + k_3 e_{3j}^2 + k_4 e_{4j}^2 = \sum_{a=1}^{4} k_a e_{ai}^2, 1 \leq i \leq 3, 1 \leq j \leq 4\).

The network has the structure with the following input-output pair:

**Input:** \((B_{11}, B_{12}, B_{13}, B_{14}, B_{22}, B_{23}, B_{24}, B_{33}, B_{34}, B_{44})^T\)

**Output:** \((\lambda_1^{-1}, \lambda_2^{-1}, \lambda_3^{-1}, \lambda_4^{-1})^T\).
(B) 2D: six inputs are mapped to three outputs
\[ H^T WH = \begin{bmatrix} B_{11} & B_{12} & B_{14} \\ B_{22} & B_{24} & \text{sym} \\ B_{44} \end{bmatrix} \] (24)

Input: \((B_{11}, B_{12}, B_{14}, B_{22}, B_{24}, B_{44})^T\)
Output: \((\lambda_1^{-1}, \lambda_2^{-1}, \lambda_3^{-1})^T\).

Type 4. (A) 3D: ten inputs are mapped to one output. This is a type of mapping from the elements of the matrix \((H^T WH)\) to approximate WGDOP. The network has the following input-output relationship:

Input: \((B_{11}, B_{12}, B_{14}, B_{22}, B_{24}, B_{33}, B_{34}, B_{44})^T\)
Output: WGDOP.

(B) 2D: six inputs are mapped to one output
Input: \((B_{11}, B_{12}, B_{14}, B_{22}, B_{24}, B_{44})^T\)
Output: WGDOP.

Type 5. (A) 3D: sixteen inputs are mapped to four outputs. Using this type of mapping, the elements of matrix \(H\) and \(W\) are utilized to approximate the inverse of the eigenvalues without having to calculate \(H^T WH\). The network has the following mapping architecture:

Input: \((e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33}, e_{41}, e_{42}, e_{43}, k_1, k_2, k_3, k_4)^T\)
Output: \((\lambda_1^{-1}, \lambda_2^{-1}, \lambda_3^{-1}, \lambda_4^{-1})^T\).

(B) 2D: twelve inputs are mapped to three outputs
Input: \((e_{11}, e_{12}, e_{21}, e_{22}, e_{31}, e_{32}, e_{41}, e_{42}, k_1, k_2, k_3, k_4)^T\)
Output: \((\lambda_1^{-1}, \lambda_2^{-1}, \lambda_3^{-1})^T\).

Type 6. (A) 3D: sixteen inputs are mapped to one output. This architecture is proposed to train the mapping for approximating WGDOP from the elements of matrix \(H\) and \(W\). The network has the following input-output relationship:

Input: \((e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33}, e_{41}, e_{42}, e_{43}, k_1, k_2, k_3, k_4)^T\)
Output: WGDOP.

(B) 2D: twelve inputs are mapped to one output
Input: \((e_{11}, e_{12}, e_{21}, e_{22}, e_{31}, e_{32}, e_{41}, e_{42}, k_1, k_2, k_3, k_4)^T\)
Output: WGDOP.

Given a number of known input vectors and corresponding output vectors, Rprop is employed to train a network until it obtains approximate WGDOP values. After the training, the elements of matrix \(H\) and \(W\) as input data cannot only pass through the trained Rprop more quickly but also predict the WGDOP more accurately. The simulation results show that the proposed Type 5 and Type 6 need fewer hidden neurons and the number of training iterations. Thus they have much reduced computational load and are more practical. Note that all the above architectures for obtaining WGDOP are applicable regardless of the number of the location measurement units.

5. Proposed BS Selection Criterion and Location Methods Using Next Rprop

5.1. Proposed BS Selection Criterion. The proposed BS selection criterion with minimum WGDOP can be modified for application in cellular communication systems. Incorporating not only BSs geometry but also NLOS error statistics, we focus on the reciprocal of the square root of the upper bound of the NLOS errors as factors to determine the weight matrix. The measurements are divided into a number of subsets, and then the location estimates rendered by the minimum WGDOP subset can be determined accordingly. In this paper, we only consider the subsets of four measurements. Thus, the measurements could be divided into 35 \((C_4^7)\) possible subsets. As such, the number of possible subsets is reduced from 35 \((C_4^3)\) to 20 \((C_4^2)\) and the computational load can be much reduced. In this way, WGDOP is computed for 20 possible subsets and the one with the smallest WGDOP is selected.

5.2. Proposed Location Methods Using Next Rprop. Based on the above BS selection criterion, the simplest location method employs the BSs with minimum WGDOP to estimate the MS location. In general, the subsets with smaller WGDOP provide more accurate MS location results. The proposed method calculates the MS location of several subsets with the first \(N\) minimum WGDOP. The corresponding higher precision MS locations of the first \(N\) minimum WGDOP subsets are defined as the candidate points. We then employ another Rprop to estimate the final MS location. Specifically, the candidate points are fed into the next Rprop which has the following input-output mapping:

Input: \(N\) candidate points \((N = 1, 2, 3)\)
Output: true MS location.

During the training period, Rprop is trained to establish the relationship between the candidate points and the true MS location. After the training, the candidate points become the input data and are passed through the trained Rprop to predict the final MS location.

6. Simulation Results

We consider the problem of mobile location using TOA measurements and attempt to improve the performance of
the MS location estimate in 2D environments. Computer simulations are performed to investigate improvement in location accuracy. We consider a center hexagonal cell (where the serving BS resides) with six adjacent hexagonal cells of the same size, as shown in Figure 2. The serving BS, that is, BS1, is located at (0, 0). Each cell has a radius of 5 km and the MS locations are uniformly distributed in the center cell [26]. The dominant error for wireless location systems is usually due to the NLOS propagation effect. NLOS error statistics are provided and can vary significantly from one BS to another. In this paper, the NLOS propagation model is based on the uniformly distributed noise model [27], in which the TOA NLOS errors from all the BSs are different and assumed to be uniformly distributed over (0, Ui), for i = 1, 2,..., 7, where Ui is an upper bound. The specific variables are chosen as follows: U1 = 200 m, U2 = 400 m, U3 = 350 m, U4 = 700 m, U5 = 300 m, U6 = 500 m, and U7 = 350 m. The reciprocal of the square root of an upper bound of the NLOS errors is set to be diagonal elements of the weight matrix W.

In the simulation, we have only considered single hidden layer, which is most commonly used. Figure 3 shows how the converged WGDOP residual varies as the number of training iterations (epochs) increases. Here, WGDOP residual is defined to be the difference between the actual WGDOP and the estimated WGDOP. Observe that WGDOP residual decreases as the number of epochs increases. At the beginning of the training period, the error is decreasing rapidly. When the number of epochs increases to more than 2000, this reduction slows down. This suggests that Rprop offers much faster convergence, compared to the traditional BPNN [22].

In the simulation, we have only considered single hidden layer, which is most commonly used. Figure 3 shows how the converged WGDOP residual varies as the number of training iterations (epochs) increases. Here, WGDOP residual is defined to be the difference between the actual WGDOP and the estimated WGDOP. Observe that WGDOP residual decreases as the number of epochs increases. At the beginning of the training period, the error is decreasing rapidly. When the number of epochs increases to more than 2000, this reduction slows down. This suggests that Rprop offers much faster convergence, compared to the traditional BPNN [22].

The number of hidden neurons can also be critical here. When there are too few hidden neurons, a bigger error may occur. On the other hand, a larger number of hidden neurons can slow down the convergence. Some general rules for determining the number of hidden neurons are (i) half of the sum of the input neurons and the output neurons; (ii) the same as that of the number of input layer’s neurons; (iii) two times the number of input neurons plus one; (iv) three times the number of input neurons plus one. Figure 4 shows the results of WGDOP residual for various numbers of hidden neurons. The hidden layer neurons with (2p + 1) provide reasonably accurate results.

Based on the optimized neural network structure stated above, the Rprop algorithm can be applied to predict WGDOP value after the training period. Figures 5 and 6 show the mean-square error (MSE) of WGDOP over time when using BPNN and Rprop, respectively. So Rprop method can offer much faster convergence time. Rprop algorithm has been used to train the neural networks and was shown to converge faster than BPNN. The proposed Rprop algorithm with six types can yield smaller MSE with very short time consuming.

From Figures 7 and 8, one can see the cumulative distribution function (CDF) curves for the six types of mapping architectures based on BPNN and Rprop. The quantity of epochs of Rprop is fewer than the traditional BPNN.
Rprop-based Type 5 with 1,000 epochs gives a superior performance than BPNN-based Types 1, 3, and 5 with 2,000 epochs. We have found that when there are only 500 pieces of input-output patterns as training data, the proposed Rprop-based Type 6 still works better than BPNN-based Types 2, 4, and 6 with 2,000 epochs. The results obtained are very promising and confirm the quality of the Rprop algorithm with respect to both convergence time and the number of epochs.

We can observe that Rprop-based method is better than BPNN-based one. The Rprop algorithm derives more accurate estimations of average WGDOP residual than BPNN with a learning rate of 0.01. The superior performance for the Rprop algorithm has been demonstrated by comparing the average WGDOP residual. We can find that the WGDOP precision of Rprop is better than BPNN. Therefore, we apply Rprop and propose an algorithm to estimate WGDOP in this paper.

Figure 9 shows CDFs of the WGDOP residual for the six types of mapping architectures with \((2p + 1)\) hidden neurons after 2000 training iterations, where \(p\) is the number of input neurons. WGDOP is equal to the square root of the sum of \(\lambda_i^{-1}, i = 1, 2, 3\), which are the outputs of three-output architectures. We can see that the one-output architectures approximate WGDOP with much better accuracy than those of the three-output architectures. The Type 1 mapping architecture predicts the eigenvalues inverse and then obtains WGDOP value with poor accuracy. The results show that the proposed Type 5 yields the best performance among the three-output architectures and the proposed Type 6 provides much better accuracy than all the other one-output architectures. The proposed Type 6 with \(p\) hidden neurons and 500 epochs renders superior performance to other architectures, such as those with \((2p + 1)\) neurons.
and 2000 epochs. In order to minimize the computational load, we used the proposed Type 6 with the aforementioned parameters in subsequent simulations because it offers satisfactory prediction performance.

With the minimum WGDOP algorithm, MS location can be estimated by the linear lines of position algorithm (LOP) [28], distance-weighted method, and threshold method which we have proposed in [29, 30]. Figure 10 shows the CDF of the average location error of these methods with different subsets. From simulation results, using a subset of the serving BS in combination with three other BSs yields comparable performance to using a subset of selecting four measurements from seven BSs. The former has many fewer choices of subsets, that is, \( \binom{6}{3} \) instead of \( \binom{7}{4} \); thus, it requires greatly reduced computation. With the proposed BSs selection criterion, Rprop-based WGDOP approximation method and matrix inversion method provide nearly identical MS location estimation.

One can see in Figure 11 that four randomly selected BSs with poor geometry yield extremely bad location estimation, and the location accuracy can be critically affected by the relative geometry between BSs and MS. In contrast, our proposed BS selection criterion always provides much better location estimation than the random subsets of four BSs chosen from seven BSs. The simulation results show that the positioning precision of using all seven BSs is only slightly better than that of the proposed BSs selection criterion. Selecting four BSs with the best geometry is generally sufficient to render substantial decrease in the positioning error.

To enhance the performance of location estimation, we apply the first \( N (N = 1, 2, 3) \) higher precision MS location with minimum WGDOP to next Rprop. Based on the \((2p+1)\) hidden neurons and 500 training iterations structure, we use only one candidate point \( (N = 1) \) and still achieved better performance as shown in Figure 12. Higher accuracy of MS location is obtained when more candidate points are taken into consideration. Our simulations show that the proposed location methods with just two candidate points can even outperform those of using all seven BSs. Therefore, it is
by using all seven BSs with matrix inversion. In essence, the proposed methods can make effective performance improvement of MS location estimation. The proposed algorithms can be applicable to all positioning techniques.

References


