

Soft Computing: Fuzzy Logic Approach in Wireless Sensors Networks

M. Vinoth Kumar¹, Dr. T. Lalitha^{2*}

¹Department of Computer Science and Engineering, AVS Engineering College, Salem, India

²Sona College of Technology, Salem, India

Email: ^{*}lalithasrilekha31@gmail.com

Received 14 March 2016; accepted 12 April 2016; published 7 June 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The recent growth of communication and sensor technology results in the enlargement of a new attractive and challenging area-wireless sensor networks (WSNs). A network comprising of several minute wireless sensor nodes which are organized in a dense manner is called as a Wireless Sensor Network (WSN). Every node estimates the state of its surroundings in this network. The estimated results are then converted into the signal form in order to determine the features related to this technique after the processing of the signals. It's high computational environment with limited and controlled broadcast range, processing, as well as limited energy. The embedded soft computing approach in wireless sensor networks is suggested. This approach means a grouping of embedded fuzzy logic and neural networks models for information processing in complex environment with unsure, rough, fuzzy measuring data. It is generalization of soft computing concept for the embedded, distributed, adaptive systems.

Keywords

Embedded Fuzzy Logic, Data Fusion, Clusterization, Aggregation, Fuzzy Distributed Knowledge Base, Sensor

1. Introduction

The knowledge of fuzzy and neuro-fuzzy systems in WSN uses soft computing approach to increase performance of wireless sensors networks (WSN) and to make them smarter. WSN is one of the most gifted technologies of the 21st century. The prototype of WSN node is a software hardware platform for operation of several specialized sensors on the base on an self-governing wireless controller.

As shown in **Figure 1**, WSN consists of a large number of tiny devices, which are deployed in real environment

*Corresponding author.

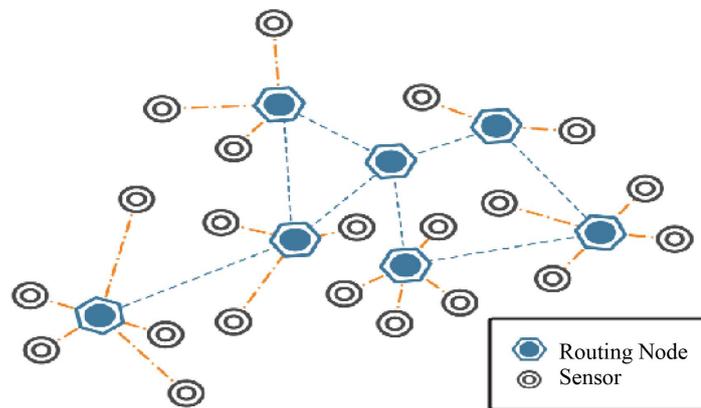


Figure 1. Common scenario for wireless sensor networks.

and function as a united network. To give sensor nodes [1] with an option of self organization the particular software was designed together with IDE for application development. The specialized software implements the possibilities of communication, routing and application support for WSN.

The rising of WSN performance means preliminary giving out of raw data, data fusion, clusterization and aggregation. Intelligent WSN provides also distributive decision-making and queries processing, knowledge-based routing and power consumption. Methods of decision-making and information processing based on pictogram models of classic artificial intelligence are too complex for embedded realization in WSN due to limited communication and power resources of WSN. Only few companies in the world solve this trouble by embedded soft computing approaches. These approaches for WSN mean a grouping of embedded fuzzy logic and neural networks models for information processing in complex environment with uncertain, vague, fuzzy measuring data. It is generalization of soft computing concept for the embedded, distributed, adaptive systems. These approaches were suggested by Russian scientists in 1997 [2], when hybridization of soft computing and mathematical statistics was used to process the results of varied measurements for environment monitoring applications. The first realization of these approaches was made in 2004. The key part of our embedded soft computing and soft computing approaches is Smart Node (SN) model for WSN [3]. The core of SN is Fuzzy Engine, which consists of three modules: knowledge base (a set of fuzzy production rules), fuzzification and defuzzification modules, which change numerical measurements in linguistic form and vice-versa. The output of SN can estimate of any function of input parameters, e.g. when it is not possible or difficult to measure certain parameter, it can be computed by SN with the use of special rules from knowledge base. Similarly the special rules can be created for data fusion, clusterization, aggregation, routing and power consumption. The knowledge base of SN can be created as a result of knowledge acquisition from expert or by supervised neural network learning. Application knowledge in nodes can significantly improve the resource and energy efficiency, for example by application-specific data caching and aggregation in intermediate node SN is realized surrounded by MeshNetics platform [4].

MeshNetics is a family of software components, algorithms, hardware designs and solutions that allow next generation M2M applications [3]. By adding expert system to monitoring, calculating, tracking applications, it enables new generation of solutions optimized for the needs of end-users. With MeshNetics™ businesses profit from reduced pricing structure, life-cycle time and enhanced competitiveness of their new and existing products. MeshNetics™ enables expert remote monitoring and control of a wide range of processes, assets, systems, and facilities. Built-in expert system with advanced algorithms, neural networks and fuzzy logic creates a new generation of WSN's and frees businesses from being trapped into costly and complex wired "static" systems and enhances the possibilities by listening actively to your surroundings. Built-in expert system on the base of SN, that allows to support hybrid distributive expert system on the nodes of WSN, which can understand, together with data collection and communication, a large class of existing algorithms of data fusion and aggregation. Meshnetics SN is the node of Meshnetics platform, which includes Smart Engine. Smart Engine is given by a number of its parameters, *i.e.* rules patterns, variables, terms, membership functions, triangular norms. These parameters can convey across WSN. These transmissions can be defined by user, or by Smart Node (SN) itself. E.g., in goal tracking process these transmissions can be realized by SN in dependence of mutual smart node and goal po-

sitions.

2. Challenges Deployment of WSN

There are some general challenges that are faced in numerous WSNs applications.

- Hardware constraint

Since WSNs rely upon battery based mostly power devices; power supplier is that the most important half within the device nodes. The less energy consumption devices in WSN are the leading economical and lasting WSN. The feature of device nodes; like the process capabilities and storage capacity; also will have an outcome on the performance and life time of WSN as they'll increase the power consumption and knowledge redundancy. The size, processing, value and therefore the amount of the sensors within the applied environment ought to be taken in thought whereas we tend to enlarge WSNs.

- Power consumption

The drawback of power resources in WSNs contrariwise the high energy consumption direct the researchers attentions to power conservation and power organization approaches that may successfully prolong the WSN life. Deployment: A WSN is associate infrastructure less at random deployed network encompass small autonomously distributed sensors. The network readying will be densely by a huge variety of sensors in practical space of application or distributed network with many and limited variety of sensors. Communication in WSNs is achieved by single or multi number of hops between sensors. The importance of application yet because of the value of deployment controls the category of WSN readying.

- Scalability

WSNs ought to be prepared to support form of routing protocols, huge nodes number and wide space of application yet because the frequent will increase of network expansion. The size of performance and employment of WSN shouldn't be predictable during the initial network style stage.

- Flexibility

Thanks to the wide numerous of WSN application, yet because the network limitation and inadequacy of resources, some kind of elasticity are required like completely different network readying method and topologies, routing protocols, power management process and then on. Reliability: A WSN ought to be ready to adapt and manage the corruption of the network in case of node breakdown. The practicality and performance to WSNs shouldn't be affected negatively. Some error tolerance techniques guarantee responsibility in WSNs.

- Connectivity

Maintain property among all device nodes through the network life time may be a horribly difficult issue. The significance of every device node yet because the meaning of perceived knowledge and routing route that every device might take urges the network to preserve the lifetime of every node. Some sleep modes will be experienced by some nodes in order to cut back the speed of harvested energy. Lifetime: The longevity and coverage of the WSN ought to be warranted. The main emphasis is to prolong the network life. Device nodes are finite life time devices as they are battery high-powered. Some adapting mechanisms like power management method and accommodative routing procedure are wont to overcome the restricted resources efficiently and to confirm the utmost network life.

3. Soft Computing Techniques

The complementarity of FL, NC, GC, and PR has an important consequence: in many cases a problem can be solved most effectively by using FL, NC, GC and PR in combination rather than exclusively. A striking example of a particularly effective combination is what has come to be known as "neuro fuzzy systems". Such systems are becoming increasingly visible as consumer products ranging from air conditioners and washing machines to photocopiers and camcorders. Less visible but perhaps even more important are neuro fuzzy systems in industrial applications. What is particularly significant is that in both consumer products and industrial systems, the employment of soft computing techniques leads to systems which have high MIQ (Machine Intelligence Quotient). In large measure, it is the high MIQ of SC-based systems that accounts for the rapid growth in the number and variety of applications of soft computing.

The conceptual structure of soft computing suggests that students should be trained not just in fuzzy logic, neuro computing, genetic programming, or probabilistic reasoning but in all of the associated methodologies, though not necessarily to the same degree.

Previous Localization Methods

Bulusu [5] *et al.* proposed a range-free, proximity-based, and coarse-grained localization algorithm. In this method, the anchor nodes broadcast their position (X_i, Y_i) and each sensor node computes its position as a centroid of the positions of all the connected anchor nodes to itself by

$$(X_{\text{est}}, Y_{\text{est}}) = \left(\frac{X_1 + \dots + X_n}{N}, \frac{Y_1 + \dots + Y_n}{N} \right) \quad (1)$$

where $(X_{\text{est}}, Y_{\text{est}})$ stand for the estimated position of the sensor node and N is the number of the connected anchor nodes to the sensor node. This method is simple and economic but exhibits large amount of error.

Kim and Kwon [6] proposed an improved version method, secure nodes are weighted according to their proximity to the sensor nodes, and each sensor node computes its location by

$$(X_{\text{est}}, Y_{\text{est}}) = \left(\frac{w_1 \cdot X_1 + \dots + w_n \cdot X_n}{\sum_{i=1}^n w_i}, \frac{w_1 \cdot Y_1 + \dots + w_n \cdot Y_n}{\sum_{i=1}^n w_i} \right) \quad (2)$$

This method has the weakness that the choice of the weights (w_1, w_2, \dots, w_n) , is very heuristic, and the performance highly depends on the design of the weights.

4. Importance of Soft Computing

Soft computing is a promising problem solving technology and, in particular, it is suitable for unsure and nonlinear problems. Fuzzy Logic Systems (FLS) produce acceptable but definite output in response to incomplete, ambiguous, distorted, or inaccurate (fuzzy) input. Fuzzy Logic (FL) is a method of reasoning that resembles human reasoning. The approach of FL imitates the way of decision making in humans that involves all intermediate possibilities between digital values YES and NO.

4.1. Fuzzy Logic Systems Architecture

It has four main parts as shown.

- Fuzzification Module-It transforms the system inputs [2], which are crisp numbers, into fuzzy sets. It splits the input signal into five steps such as:

LP	X is Large Positive
MP	X is Medium Positive
S	X is Small
MN	X is Medium Negative
LN	X is Large Negative

- Knowledge Base-It stores IF-THEN rules provided by experts.
- Inference Engine-It simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN rules.
- Defuzzification Module-It transforms the fuzzy set obtained by the inference engine into a crisp value. The membership functions work on fuzzy sets of variables.

Membership Function

Membership functions allow you to quantify linguistic term and represent a fuzzy set graphically.

As shown in **Figure 2**, A membership function for a fuzzy set A on the universe of discourse X is defined as $\mu_A: X \rightarrow [0, 1]$.

Here, each element of X is mapped to a value between 0 and 1. It is called membership value or degree of membership. It quantifies the degree of membership of the element in X to the fuzzy set A .

- x axis represents the universe of discourse.
- y axis represents the degrees of membership in the $[0, 1]$ interval.

There can be multiple membership functions applicable to fuzzify a numerical value. Simple membership functions are used as use of complex functions does not add more precision in the output.

All membership functions [3] for LP, MP, S, MN, and LN are shown as below-

In **Figure 3**, the triangular membership function shapes are most common among various other membership function shapes such as trapezoidal, singleton, and Gaussian.

Here, the input to 5-level fuzzifier varies from -10 volts to +10 volts. Hence the corresponding output also changes. Soft computing can be measured as collaboration between the FLS, NN, GA and other AI (artificial intelligence) linked fields. In this paper, soft computing techniques play a vital role in the future localization methods.

4.2. Fuzzy Logic System

The fuzzy logic system (FLS) [1] is an inference scheme which mimics the human thoughts and its basic configuration consists of a fuzzifier, some fuzzy IF-THEN rules, a fuzzy inference engine and a defuzzifier, as shown in **Figure 4**.

A fuzzy rule is written as the following statement:

$$R^i : \text{IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and } \dots x_n \text{ is } A_n^i \text{ THEN } y \text{ is } y^i \tag{3}$$

where R^i ($i = 1, 2, \dots, l$) denotes the i th implication; x_j ($j = 1, 2, \dots, n$) is input variables of the FLS; y_i is a singleton; A_j^i is the fuzzy membership functions, which can represent the uncertainty in the reasoning. When we employ the product inference, center-average and singleton fuzzifier, the output of the fuzzy system for an input $X = (x_1, x_2, \dots, x_n)$ can be expressed as

$$y = \frac{\sum_{i=1}^N \alpha_i y_i}{\sum_{i=1}^N \alpha_i} \tag{4}$$

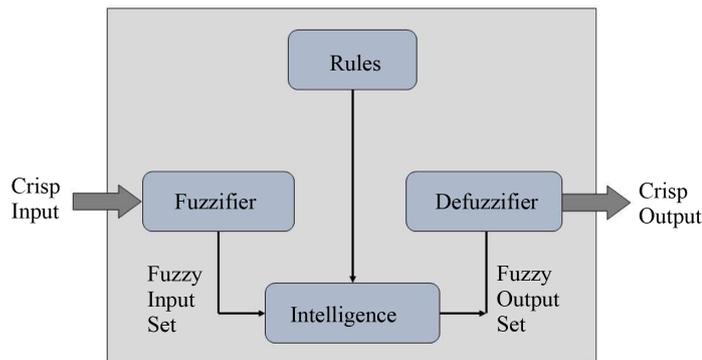


Figure 2. Fuzzy logic architecture.

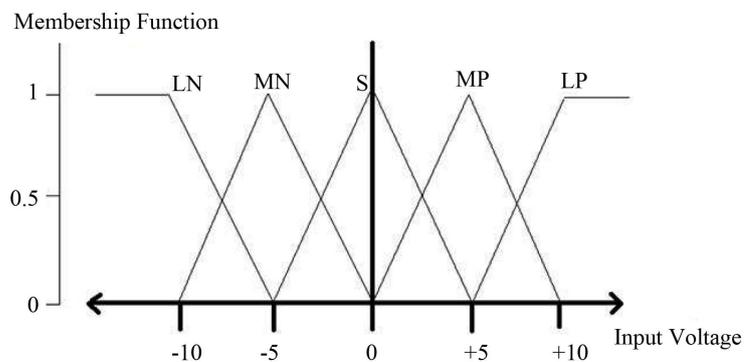


Figure 3. Membership function.

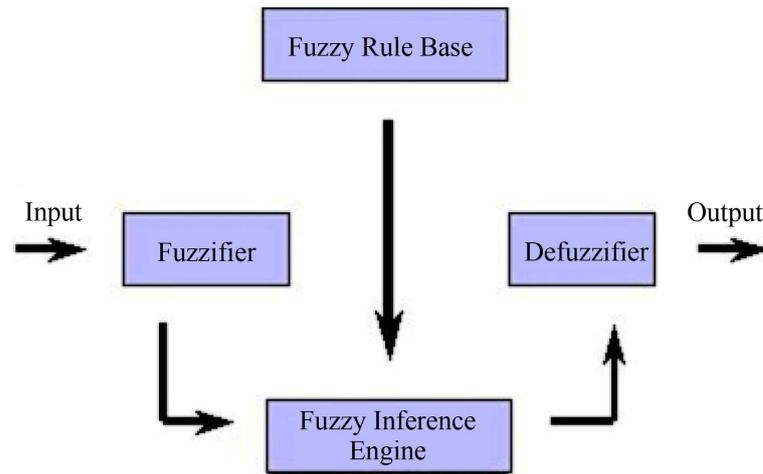


Figure 4. An example of the fuzzy logic system (FLS).

where α_i implies the overall truth value of the premise of the i th implication, and are computed as

$$\alpha_i = \prod_{j=1}^n A_j^i(x_i) \quad (5)$$

4.3. Genetic Algorithm

Genetic algorithms (GAs) are numerical optimization algorithms inspired from genetics and have been applied to a wide range of problems [2]. GA typically maintains a population of persons that symbolize a set of candidate key for the considered problem. The kindness of each candidate solution is evaluated based on its strength to the objectives, and the population develops by selection, crossover, and mutation. In the selection process, some individuals are chosen to be copied into a tentative next population. The number of copies of each individual in the next generation is proportional to its relative fitness value. The promising individuals are therefore more likely to be selected for the next generation. The selected individuals are varied to search for a global optimal solution using mutation and crossover. GA is simple yet provides an adaptive and robust optimization methodology.

4.4. Neural Network

Neural networks (NNs) imitate the human brain to achieve intelligent tasks [3]. They can represent complicated relationships between input and output variables, and acquire knowledge about these relationships directly from the data. The type of NN used in this paper is the multilayer perceptron (MLP) and consists of an input layer, a nonlinear hidden layer and a linear output layer, as shown in **Figure 5**.

In **Figure 5**, when the numbers of neuron units of input, hidden and output layers are denoted by n , p , and m , respectively, the NN yields the output

$$P(\omega_k|X) \approx y_k = f \left\{ \sum_{j=1}^p w_{jk} f \left(\sum_{i=1}^n v_{ij} x_i \right) \right\} \quad \text{for } k = 1, 2, \dots, m \quad (6)$$

where v_{ij} s are input-to-hidden layer weights; w_{jk} s are hidden-to-output weights; and f is an activation function.

5. Result

Simulation

Setup and Performance Measure

First, we carry out the computer simulation to prove the efficiency of the planned intelligent localization methods. The algorithms are coded in Matlab. In this simulation, we consider a 100×100 m² region. One hundred and twenty one anchor nodes are placed regularly with each one 10 m apart and 60 sensor nodes are placed arbitrarily as shown in **Figure 6**.

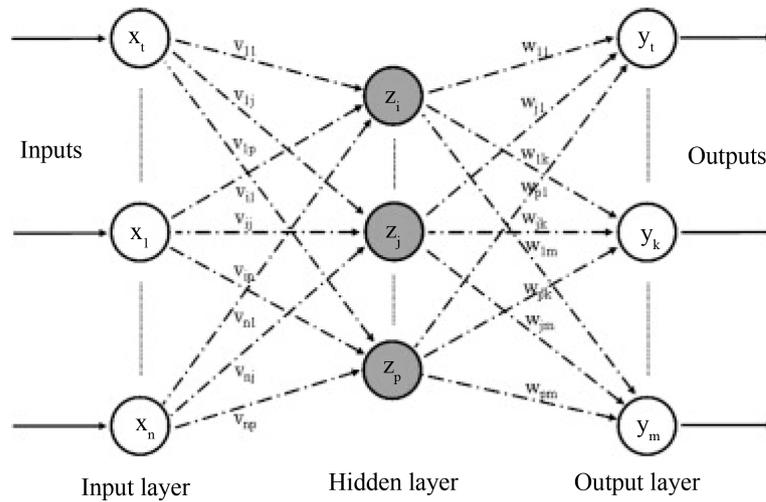


Figure 5. An example of the MLP neural networks.

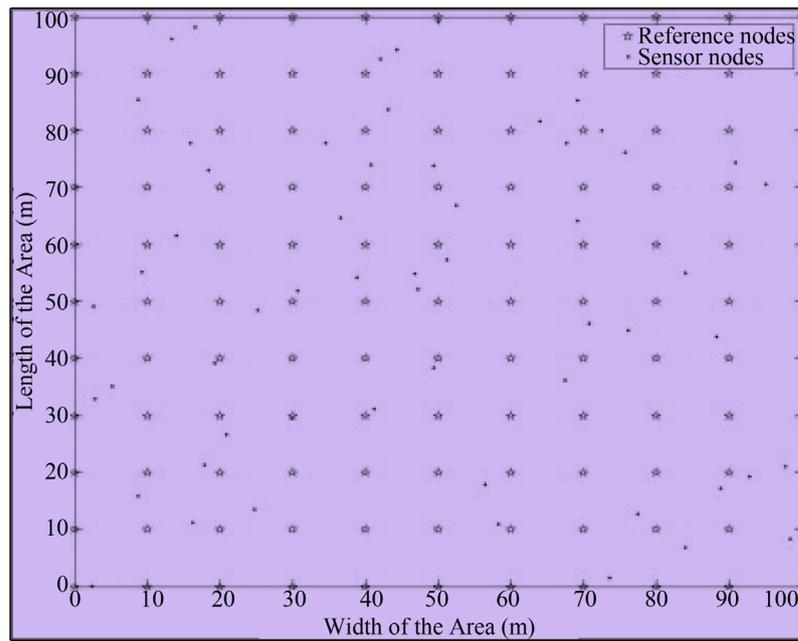


Figure 6. Node distribution.

The transmission range of all anchor nodes is assumed 8.94 m. A sensor node can communicate with adjacent anchor nodes if its distance from the anchor node is smaller than the transmission range. For the simulation, we used the following RSS model which also takes into account noise

$$R_{ij} = (kd_{ij}^{-\alpha}) + (AWGN * Var.) \tag{7}$$

where R_{ij} is the RSS value between the i th sensor node and the j th adjacent anchor node, k is a constant which takes into account carrier frequency and transmitted power, d_{ij} is the distance between the i th sensor node and the j th adjacent anchor node and α is the attenuation exponent. Here, we use $k = 50$ and $\alpha = -1$.

To evaluate the proposed schemes, we use the following two performance indices.

1) *Location error*: The distance between the estimated position and the actual position of sensor node.

$$\text{Location error} = \sqrt{(X_{\text{est}} - X_a)^2 + (Y_{\text{est}} - Y_a)^2} \tag{8}$$

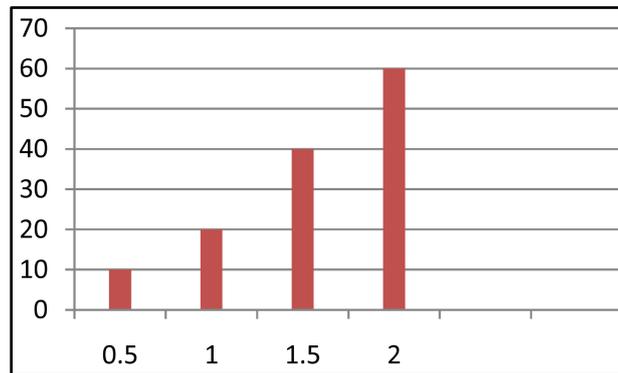


Figure 7. Simulation result of the proposed individual method by FLS and GA error result.

2) *Average location error*: The average distance between the estimated position and the actual position of all sensor nodes.

$$\text{Average location error} = \frac{\sum \sqrt{(X_{\text{est}} - X_a)^2 + (Y_{\text{est}} - Y_a)^2}}{\text{number of sensor nodes}} \quad (9)$$

Soft Computing is the synthesis of methodologies that were considered to model and enable solutions to real world troubles, which are not modeled or too hard to model, scientifically. Soft computing is a collection of methodologies that works synergistically and provides, in one form or another, elastic information processing capability for conduct real-life ambiguous situations.

As shown in **Figure 7**, Genetic Algorithms (GAs) is a soft computing approach. Ga is general-purpose search algorithms, which use values stimulated by natural genetics to evolve solutions to problems. As one can guess, genetic algorithms are inspired by Darwin's theory about development. They have been successfully applied to a large number of scientific and engineering problems, such as optimization, machine learning, automatic programming, transportation problems, adaptive control etc.

6. Conclusion

The work presents the way to evolve the fuzzy if/then rules using GA for planned intelligent localization problem. Sensor location estimation in Wireless Sensor Networks is a non-deterministic problem in actual conditions which can be modeled by FLS. FLS can be a general framework for combining fuzzy logic with probability theory and hence a better handling of cases where both sources exist. In this research, uncertainty in the localization problem is modeled by FLS. The paper simulates and examines localization with FLS in a dynamic environment and compares the behavior of these methods with others such as NLP and NN. Also in future the system can also be implemented for many other applications and problems involving wide range of classification.

References

- [1] Sugeno, M. and Kang, G.T. (1986) Fuzzy Modeling and Control of Multilayer Incinerator. *Fuzzy Sets and Systems*, **18**, 329-346. [http://dx.doi.org/10.1016/0165-0114\(86\)90010-2](http://dx.doi.org/10.1016/0165-0114(86)90010-2)
- [2] Coley, D. (1999) An Introduction to Genetic Algorithms for Scientists and Engineers. World Scientific. <http://dx.doi.org/10.1142/3904>
- [3] Hagan, M.T., Demuth, H.B. and Beale, M. (1996) Neural Network Design. PWS Publishing Company.
- [4] Yun, S., Lee, J., Chung, W., Kim, E. and Kim, S. (2009) A Soft Computing Approach to Localization in Wireless Sensor Networks. *Expert Systems with Applications*, **36**, 7552-7556. <http://dx.doi.org/10.1016/j.eswa.2008.09.064>
- [5] Bulusu, N., Heidemann, J. and Estrin, D. (2000) GPS-Less Low Cost Outdoor Localization for Very Small Devices. *IEEE Personal Communications Magazine*, **7**, 28-34.
- [6] Kim, S.Y. and Kwon, O.H. (2005) Location Estimation Based on Edge Weights in Wireless Sensor Networks. *Journal of Korea Information and Communication Society*, **30**, 877-978.