Soft Computing Applied in Knowledge Engineering Problem Solving

Liya Ding

Inter-University Institute of Macau
NAPE, Lote 18, Rua de Londres – P, Edf. Tak Ip Plaza, R/C – 3° andar
E-mail: liya@iium.edu.mo

***Abstract*—This article discusses the application of fuzzy logic and soft computing in knowledge engineering problem solving. The discussion extends to modeling of knowledge-based systems using soft computing, and handling of uncertainty and similarity. Application examples are introduced, which range widely from control of optimization, diagnosis, prediction, to Web-based document retrieval.**

***Index Terms*— Evolutionary Computation, Fuzzy Logic, Knowledge Engineering, Neural Networks, Soft Computing.**

I. Introduction

Knowledge Engineering (KE), known as an important component of artificial intelligence (AI), is an area that mainly concentrates on activities with knowledge, including knowledge acquisition, representation, validation, inference, and explanation. Soft computing (SC), on the other hand, is an area that provides tools and methodologies for intelligent systems to be developed with the capability of handling uncertainty and imprecision, learning new knowledge and adapting themselves to a changing environment.

Though the concept of knowledge engineering was put forward in its own way in early years without the recognition of the usefulness of soft computing, soft computing methodologies, including fuzzy logic, neural networks, and evolutionary computation, have been related to one or more aspects of KE and therefore AI problems. They have done so with their particular strengths from the beginning.

There have been many remarkable works done in parallel in both KE and SC areas, but relatively less in the intersection of the two. In recent trends, many researchers of SC have applied their techniques in solving KE problems, and researchers of KE have adopted SC methodologies to enhance KE applications. We may expect increasing applications of soft computing in knowledge engineering problem solving.

The rest of this article is organized as follows. Section 2 of this article briefly describes how soft computing contributes to different aspects of knowledge engineering. Two important issues in developing knowledge-based system using soft computing are discussed in section 2. They are (a) modeling of knowledge-based system using soft computing; (b) handling of similarity. Section 3 gives examples of practical application of fuzzy logic and soft computing with knowledge engineering techniques. The conclusion is provided in section 4.

II. Soft Computing Contributes to Knowledge Engineering

A. Soft Computing

Soft computing, as pointed out by Prof. Lotfi A. Zadeh, is an emerging approach to computing which parallels the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision. Fuzzy logic, artificial neural networks, and evolutionary computation are the three principal members of soft computing [1,2].

Fuzzy logic According to its narrow sense, fuzzy logic is viewed as a generalization of the various multi-valued logics. It mainly refers to approximate reasoning, as well as knowledge representation and inference with imprecise, incomplete, uncertain, or partially true information. According to the broad sense, fuzzy logic includes all the research efforts related to fuzzy inference systems (or fuzzy systems).

It is generally agreed that human knowledge includes imprecision, uncertainty, and incompleteness in nature, because the human brain interprets imprecise and incomplete sensory information provided by perceptive organs. Instead of simple rejection of the ambiguity, fuzzy set theory, as an extension of set theory, offers a systematic calculus to deal with such information. It performs numerical computation by using linguistic labels stipulated by membership functions. With fuzzy sets, human knowledge described in words can be represented and hence processed in computer. Fuzzy logic, in its narrow sense, offers the possibility of inference with uncertainty and imprecision. Together with fuzzy set theory, it provides the basis of fuzzy inference systems.

Fuzzy knowledge representation and approximate reasoning have greatly extended the ability of the traditional rule-based system. However, it lacks the adaptability to deal with a changing environment and assumes the availability of well-structured knowledge for the problem domain. Thus, people incorporate learning concepts in fuzzy inference systems.

One important way of materializing learning in fuzzy inference systems is using neural networks.

Neural Networks A neural network system is a continuous-time nonlinear dynamic system. It uses connectionist architectures to mimic human brain mechanisms for intelligent behavior. Such connectionism replaces symbolically structured representation with distributed representation in the form of weights between a massive set of interconnected processing units. The weights are modified through a certain learning procedure so that the neural network system can be expected to improve its performance progressively in a specific environment.

Neural networks are good in fault-tolerance, and can learn from training data provided in non-structured and non-labeled form. However, as compared with fuzzy inference systems, the knowledge learned in a neural network system is usually non-transparent and hard to explain. Many researchers have put efforts on rule extraction from neural networks and rule generation using neural networks. Those extracted or generated rules can then be used to develop fuzzy inference systems with necessary and possible fine tuning.

Evolutionary Computation Fuzzy logic offers knowledge representation and inference mechanism for knowledge processing with imprecision and incompleteness. Neural networks materialize learning and adaptation capability for intelligent systems. While evolutionary computation provides the capacity for population-based systematic random search and optimization.

A `best' solution may always be expected for many AI applications. The use of heuristic search techniques, therefore, forms an important part of application of intelligent systems. However, in reality it is not always possible to get such an optimal solution when the search space is too large for an exhaustive search and at the same time is too difficult to reduce. Genetic algorithm (GA) is a usable technique to perform more efficient search techniques to find less-than-optimum solutions.

B. Soft Computing Contributes to Knowledge Engineering

Knowledge engineering is a discipline devoted to integrating human knowledge in computer systems, or in other words, to building knowledge-based systems. It can be viewed from both the narrow or wider perspectives. According to the narrow perspective, knowledge engineering deals with knowledge acquisition (also referred to as knowledge elicitation), representation, validation, inference, and explanation. Alternatively, according to the wider perspective the term describes the entire process of development and maintenance of knowledge-based systems. In both cases *knowledge* plays the key role.

Knowledge engineering, especially the knowledge acquisition practice, involves the cooperation of human experts in that domain who work with the knowledge engineer to codify and to make explicit the rules (or other form of knowledge) that a human expert uses to solve real problems. Since the construction of knowledge base needs human knowledge in a direct or an indirect way, an important issue in the design of knowledge-based systems is how to equip them with human knowledge that often appears to be *uncertain*, *imprecise*, and *incomplete* to some degree. Some of the contributions of soft computing to knowledge engineering can be found in the following aspects.

Knowledge Representation Fuzzy logic can be used to represent imprecise and incomplete knowledge described in words. On the other hand, knowledge based neural networks offer a connectionist way of knowledge representation with the learning ability of neural networks.

Knowledge Acquisition When the information or data obtained as domain knowledge is less structured and summarized, neural networks can be employed for learning. A trained neural network can be viewed as a form of knowledge representation and possible rule extraction may be applied then to obtain fuzzy rules from the neural network. Some clustering techniques can also be used with fuzzy logic to help fuzzy rule extraction. Genetic algorithms can help search for more accurate fuzzy rules and fuzzy membership functions, or fine tune fuzzy systems.

Knowledge-based Inference In broad sense, both fuzzy inference systems and neural network systems offer knowledge-based inference. In fuzzy inference systems, inference is executed by using fuzzy rules, fuzzy relations, and fuzzy sets within the frame of fuzzy logic. While in neural network systems, the inference results are determined by the inference algorithms based on the learned knowledge in the neural networks. Genetic algorithms can be used to find a better neural network configuration.

Modeling and Developing Knowledge-based Systems Neuro-fuzzy modeling is a pivotal technique in soft computing by incorporate neural network learning concepts in fuzzy inference systems. Hybrid systems provide more capability in developing intelligent systems with the cooperation of the SC techniques.

Knowledge Integration Knowledge integration becomes a critical issue to maximize the functionality of an intelligent, knowledge-based system when the knowledge for the specific domain exists at different levels of abstraction and completeness, or comes from various sources and is described in different forms. The cooperation of soft computing techniques offers more flexibility in dealing with such situation.

Knowledge Discovery If we can say that knowledge representation is for representing available knowledge, and knowledge acquisition is for obtaining existing but not well summarized knowledge, then we probably should say that knowledge discovery is to find out knowledge existing in more unknown form. Neural networks, with supervised or unsupervised learning approaches, can help discover knowledge from given data. Evolutionary computations, and probabilistic approaches have also been applied for similar purposes.

C. Modeling Knowledge-based System

1) Domain Knowledge with Uncertainties

Good domain knowledge is a key of success in developing a knowledge-based system. However, it is not always possible to get a complete model of the target system with sufficient domain knowledge. Soft computing offers possible solution to solve this problem.

Incompleteness of knowledge may appear at different levels, and uncertainty of application domain may be involved in various forms. This gives us a difficulty in modeling a knowledge-based system. It is often asked that which of soft computing approaches should be applied to solve a specific problem with uncertainty and incompleteness.

It is generally believed that fuzzy systems are more preferred to handle applications with uncertainty in description of knowledge, such as fuzzy or vague terms; and a neural network based system will be more powerful to learn applications with unknown factors (variables), such as financial forecasting. Good combination of neural nets, fuzzy systems, other soft computing techniques make successful hybrid systems for many applications.

2) Similarity and Satisfaction

The term ‘similarity’ has been used in a wide sense to evaluate how similar or how different between two objects. By most of the existing formal definitions of similarity, it is a symmetrical concept. That means, two descriptions “*a* is similar to *b*” and “*b* is similar to *a*” will be evaluated as the same without considering possible difference in semantics.

However, when an evaluation of similarity is used in reasoning and retrieval with certain kind of uncertainty or vagueness, the assumption of symmetrical nature is not always a prefect idea. The concept of satisfaction measure has been proposed to take into account semantic difference involved in result of fuzzy reasoning retrieval [10].

III. APPLICATION EXAMPLES

In recent trends, many researchers of SC have applied their technique in solving KE problems, and researchers of KE have adopted SC methodologies to enhance KE applications. The efforts of integrating SC and KE can be summarized to two types, though the distinction may often be unsharp. The first type may be considered as “SC tool development”. It is to develop algorithms, methods, or approaches based on soft computing concepts, and then apply the result to solve the target problem in knowledge engineering. In this case, main emphasis is often put on the development and enhancement of SC approaches or algorithms with the given application problem as test-bed. The works introduced in the edited book “A New Paradigm of Knowledge Engineering by Soft Computing” [1] are mainly falling into the first type.

The second type can be understood as “KE problem solving using SC tool”. It is to model a knowledge-based system framework based on knowledge engineering methodologies, and then seek for suitable SC technique(s) to solve the problem for partial or entire system. In this type of application, main focus is usually the KE problem solving by using existing SC techniques with necessary modification. The examples introduced below are basically of the second type.

A. Fuzzy Inference for Optimization based on Human Experience

A significant proportion of military equipment failure is due to random effects of unreliability. When the equipment fails, there is a need to restore the equipment to its normal operational condition as quickly as possible. It is necessary to stock spare components for replacement. However, the spares are usually expensive and with certain life period, so people have to optimize the amount stocked considering both availability and spares cost. Previously, an experienced logistics analyst needs to determine the amount to stock by iteratively running a simulation program which models the dynamics of usage, failure, and repair of equipment, and increasing the amount of spares until the availability target is reached. It is a very time consuming, and tedious work, and requires a lot of experience to achieve an optimal allocation. An intelligent spares optimizer [4] has been developed to help the logistics analyst automatically running the simulation program iteratively for optimization of the required availability with minimum spares cost. The system is expected to achieve following benefits:

- The reduction in the time required to arrive at a “good” solution given the targets in minimum total spares cost and maximum equipment availability;
- Less dependence on analyst’s experience, and more consistency in results;
- Less attention time of analyst required, so improvement in productivity.

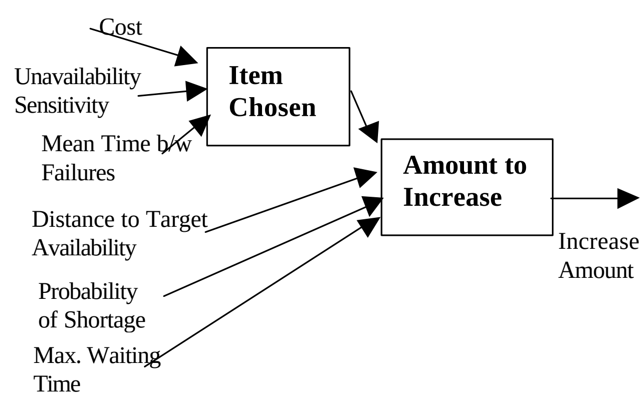

Figure 3.1 *Fuzzy Inference for Optimization of Spares*

This application can be considered as searching for optimization based on the evaluation of possible solution from the simulation tool. As the problem is highly complicated in nature, exhaustive search may not be suitable.

What an analyst uses to solve problem is basically a set of hill-climbing heuristics that tend to be imprecise and subjective. For example, the analyst may decide a *small* amount of increase for the selected item based on a *medium* distance to achieve the given target availability and the cost ceiling.

The developed system uses fuzzy inference to control which item to be increased and the amount to increase for optimization of the amount of spares given availability and cost. The inference will first determine how confident to choose an item to increase, and then determine the amount to increase. The overall inference flow can be conceptually represented as in Figure 3.1.

There are two rule blocks in the system. All the input and output variables are linguistic variables using fuzzy sets. In this way, the experience of an analyst described in semantic terms, such as *small-increase*, *large-increase*, etc., can be captured. Since numerical value of increase is expected, a defuzzification process is necessary for the inference.

Though there is still much room for further fine tuning, the test result shows that the developed prototype was able to emulate the ability of an experienced analyst in decision making.

B. Representation of Imprecise and Incomplete Knowledge for Medical Diagnosis

Traditional Chinese medicine (TCM) theory is known as one of the oldest, and the most comprehensive and powerful systems of human health care in the world. The fundamental TCM deals with balance of the Yin and Yang and believes that Yin and Yang work together to ensure harmony in the human body. Based on the TCM theory, any illness is the result of the imbalance of Yin and Yang. The TCM also views human body as a part of the universe and looks for a total eradication of the sickness. Using the theory of TCM, a TCM practitioner captures the symptoms and signs of the patient. This information will then be used to identify the patterns of disharmony to investigate the root cause of the patient's illness.

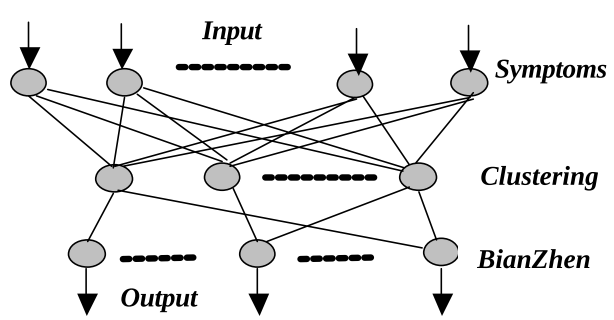

Figure 3.2 *Neural Network for TCM-based Diagnosis*

However, with much imprecision involved in its nature, the TCM system usually relies more on individual's experience and also requires longer training period to be mastered. A prototype of TCM-based Diagnosis System [3] has been developed to help this situation. It aims at following benefits:

- A reference system for doctors by providing consistent diagnosis through a systematical inference;
- A learning system for less experienced users by reducing the learning curve;
- A useful tool for patients by offering help in self-diagnosis.

In general, a diagnosis application can be a typical problem for knowledge engineering, with the domain knowledge represented in rules, or cases. However, the relationship between symptoms and illness in TCM is highly complex:

- One or more Signs and Symptoms can reflect disorder in one or more Organs;
- One or more Signs and Symptoms are required by one or more diagnostic techniques to perform a diagnosis;
- One or more Signs and Symptoms are caused by one or more causes of illness;
- One or more causes of illness can be diagnosed by one or more diagnostic techniques;
- One or more causes of illness can affect one or more Organs.

With the complexity, the imprecision, and incompleteness of the problem domain, classical knowledge-based system techniques are found insufficient and so soft computing has been selected.

The knowledge base is implemented using a neural network as shown in Figure 3.2. An RCE type of NN is adopted. The initial network is built using basic theory of Chinese medicine with the symptoms as inputs and corresponding BianZhen as outputs. Successful diagnosis records will then be kept as cases to support incremental learning.

There are three possible values for a symptom input: YES (exist), NO (not-exist), and UNKNOWN (not-sure, no-idea). By introducing the concept of fuzzy matching degree, the inference based on imprecise or incomplete information is possible. Using different values of recognition threshold for test based on a few hundreds cases, the average error rate is between 3~13%.

C. Fuzzy Information Granulation and Soft Computing Applied for Weather Forecast

At present, whenever a weather prediction is required, a human weather forecaster will give a weather forecast by monitoring the current satellite imageries and examining past satellite pictures. This is a subjective procedure that depends very much on the experience of the forecaster. The basic reasoning process of the forecaster involves extrapolation – the evolution of the weather patterns as seen on the satellite pictures for the past few days. There is a major problem faced under the current practice: whenever a short-range weather prediction is required, the human forecaster will have to spend some time in front of the displaying terminal, analyzing the current satellite image and browsing the recent ones, in order to extrapolate a weather forecast. The process is largely based on human forecaster's experience, as well as his memory of the time sequences of the satellite pictures. This causes the predicted result subjective, less accurate and inconsistent among different weather forecasts.

A decision support tool has been developed to help overcome this problem [6,7]. It automates the process of extracting cloud features from satellite images, and using neural network to generate a forecast picture. The processing flow includes a few stages.

(a) Filter image data whose pixel exceeds a threshold value. The filtered data contains cloud pixel information that is necessary in subsequent image feature extraction.

In the original satellite image, presence/absence of clouds at a particular location are determined by examining the pixel value of the weather image at the corresponding pixel position. The satellite picture is a grayscale image with pixel values ranging from 0 to 255. Based on the forecasters' experiences, images are usually segmented into three regions representing clear, intermediate or cloudy skies.

(b) Subtractive clustering is performed to identify cloud clusters.

The most prevalent and elementary unsupervised clustering method is the K-Means clustering, fuzzy C-Means clustering, a fuzzified version of K-Means. Both methods require a pre-determined number of clusters, as well as their initial cluster position, as input parameters. As it is not possible to know in advance the exact number of cloud clusters in an image for weather forecast, subtractive clustering [8] is employed, which does not require an initial number of clusters.

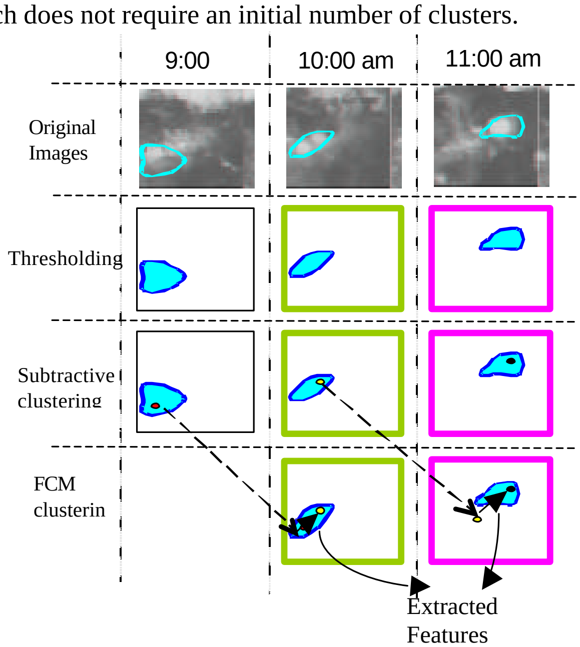

Figure 3.3 Extraction of Cloud Cluster and Velocities

(c) Fuzzy C-Means clustering algorithm is employed to identify cloud clusters using initial cluster centers from the subtractive clustering

The extracted cloud cluster information from the previous image will be used as initial seeds for fuzzy C-Means clustering on the cloud cluster of the subsequent image. This process identifies the two corresponding cloud clusters at previous and current hourly images in the sense that the later cloud cluster evolves from the earlier one.

(d) Match clustering results across consecutive hourly images.

This matching process is performed on all cloud clusters across hourly images. Cloud cluster velocities are derived from the pairing of cloud clusters.

(e) The information of cluster velocities is then fed into a neural network to predict the cluster position in the next hour.

A neural network is applied for this purpose. It has 2 layers of neurons: a radial basis layer and a special linear layer, to interpolate a 2-dimension vector field of velocities. The radial basis layer stores the cloud cluster locations, while the linear layer holds the cloud cluster velocities. For a given input location, the network outputs a vector sum of the existing cloud cluster velocities weighted on the physical proximity of the input to its neighboring clusters. It will be used as the velocity over that point.

This system has realized 1-hour ahead forecast with a significant reduction of error by 20~40%, against the persistence forecast currently used.

D. Fuzzy Sets and Soft Computing Applied for Web-based Document retrieval

The amount of information available on the Web is increasing every minute and, in certain sense, has reached unmanageable size. It becomes almost impossible for every user to read through all the available articles to search information. This causes difficulties in retrieving relevant information from the Web:

- The amount of information is too huge for a single user to manage;
- Together with relevant information, lots of unwanted information is also retrieved. It is difficult to exclude unwanted information without reading through the articles;
- Most of the existing search engines retrieve information based on search keywords. Very often, the retrieved articles may not be relevant to user's requirement.

As the result, large amount of time and resources are wasted in trying to get useful information.

A smart Internet news agent [9] has been proposed to address this problem. The proposed system is an enhancement to the commonly available search engines. Based on the input from the user, it searches specific domain related information that has been downloaded from the Web. The search conditions will be constructed based on the user input, pre-defined set of keywords as well as their weights. The respective domain experts will generate the pre-defined set of keywords. For different domain, a domain specific dictionary with a set of pre-defined keywords as well as their weights will be created. The system will offer the functions for user:

- to select a personalized "agent" from user saved set of article library;
- to rank articles in the article library based on search conditions;
- to search similar articles.

The set of keywords used for search consists of two parts. One is from domain expert, and the other part is from user's preferences. Each of the keywords is given a weight. A default weight value will be set by the system if the user has not provided it. The content of a retrieved document is represented as a histogram on corresponding keywords. The frequency of a keyword in reference document is represented as a fuzzy number with the keyword count as the central value. Similarity between user's preferences and a document is performed using fuzzy sets and soft matching. Figure 3.4 gives an example of the soft matching with 8 as the keyword count in the reference document and 6 as the keyword count in the test document. The corresponding similarity score is 0.5. The overall similarity score between a reference document and a test document will be calculated as the weighted average of all similarity scores of keyword.

In general, user's preferences for retrieval may not be fully described by a small set of keywords, in other words, the similarity model built based on keywords is incomplete. In order to solve this problem, the system also receives from user a feedback on the relevance of retrieved document, which will be associated to the article as its satisfaction score. Articles' keyword profile and their satisfaction scores will form user's preference patterns. With enough such patterns, a neural network can be trained to capture the complex "mapping" relationship between keywords and satisfaction, and so to predict satisfaction score with given keyword counts of a document.

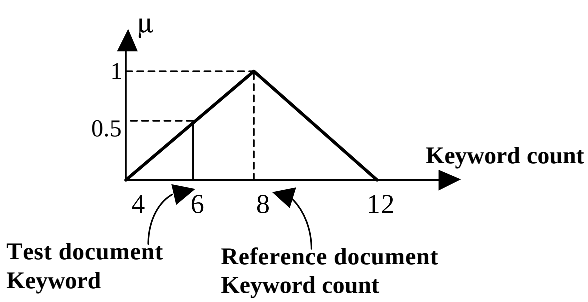

Figure 3.4 *Soft Matching for Keyword Count*

IV. CONCLUSION

The contributions of soft computing to knowledge engineering were explained. Modeling of knowledge-based system using soft computing, and the handling of uncertainty and similarity were briefly discussed. Four application examples of soft computing in knowledge engineering problem solving were introduced.

Among the four examples, the first application (1998) is basically a fuzzy rule based inference system that incorporates with imprecise human knowledge; the second application (1999) handles imprecise knowledge and allows reasoning based on incomplete information, moreover it also pays more attention on learning capability; the third application (2000) uses soft computing techniques in a wider sense to construct a hybrid architecture for information processing; the forth one, as an on-going project in 2001, represents the recent trend of fuzzy logic and soft computing applied in Web-based document processing. Although they were not purposely selected, it is very interesting that from the four examples we may trace out an epitome of the evolution of application of fuzzy logic and soft computing:

** Fuzzy rule based inference ->*
** Reasoning and learning based on uncertainty ->*
** Soft computing for information granulation ->*
** Fuzzy logic and SC applied for Web-based information processing.*

Both soft computing and knowledge engineering are rapidly developing and constantly evolving areas. More and more new techniques and applications of SC and KE are being proposed. The results achieved so far have already established a good foundation in building more "intelligent" machines in future, which will contribute greatly to our daily life.

ACKNOWLEDGMENT

The author thanks her students of Knowledge Engineering, Master of Technology in the Institute of Systems Science, National University of Singapore, for their efforts and excellent work that made those applications of SC in KE successful.

REFERENCES

[1] Lotfi. A. Zadeh, "Fuzzy logic, neural networks and soft computing", Comm., ACM, Vol. 37, No. 3, pp.77-84, 1994.
[2] J.-S. R. Jan, C.-T, Sun & E. Mizutani, "Neuro-Fuzzy and Soft Computing", Prentice-Hall, Inc., 1997.
[3] Liya Ding (Ed.), A New Paradigm of Knowledge Engineering by Soft Computing, World Scientific, 2001.
[4] Siew-Hiang Teo, Peck-Chuan Tan, U Sann Maung, & Kok-Leong Chia, "Spares Optimiser" (project report), Institute of Systems Science, National University of Singapore, 1999.
[5] Leonard Cheok, Wang Yong, Zhang Li & Kwok-Wai Lee, "Traditional Chinese Medicine Diagnosis System" (project report), Institute of Systems Science, National University of Singapore, 2000.
[6] Chien-Wan Tham & Sion-Hui Tian, "Weather Satellite Imageries Forecasting System" (project report), Institute of Systems Science, National University of Singapore, 2001.
[7] Chien-Wan Tham, Sion-Hui Tian & Liya Ding, "Weather Forecasting System based on Satellite Imageries Using Neuro-Fuzzy Techniques", 2002 AFSS International Conference on Fuzzy Systems.
[8] Chiu, S. L., "Fuzzy Model Identification based on Cluster Estimation", Journal of Intelligent and Fuzzy Systems, Vol. 2, No. 3, pp. 267–278, 1994.
[9] Rajamohan Prasad, Chee-Yuen Au, & James Chien, "Smart Internet News Agent for X-Domains" (project report), Institute of Systems Science, National University of Singapore, 2001.
[10] Liya Ding & Masao Mukaidono, "Satisfaction Measure for result in Fuzzy Reasoning and Retrieval", Joint 9th IFSA World Congress and 20th NAFIPS International Conference (IFSA/NAFIPS'2001), Vancouver, 2001.