Evolutionary Fuzzy Neural Inference System for Decision Making in Geotechnical Engineering

Min-Yuan Cheng[1]; Hsing-Chih Tsai[2]; Chien-Ho Ko[3]; and Wen-Te Chang[4]

Abstract: Problems in geotechnical engineering are full of uncertain, vague, and incomplete information. In most instances, successfully solving such problems depends on experts' knowledge and experience. The primary object of this research was to develop an evolutionary fuzzy neural inference system (EFNIS) to imitate the decision-making processes in the human brain in order to facilitate geotechnical expert decision making. First, an evolutionary fuzzy neural inference model (EFNIM) was constructed by combining the genetic algorithm (GA), fuzzy logic (FL), and neural network (NN). In the proposed model, GA is primarily concerned with optimizing parameters required in the fuzzy neural network; FL with imprecision and approximate reasoning; and NN with learning and curve fitting. This research then integrates the EFNIM with an object-oriented computer technique to develop an EFNIS. Finally, the potential to apply the proposed system to practical geotechnical decision making is validated using two real problems, namely estimating slurry wall duration and selecting retaining wall construction methods.

DOI: 10.1061/(ASCE)0887-3801(2008)22:4(272)

CE Database subject headings: Artificial intelligence; Algorithms; Neural networks; Decision making; Fuzzy sets; Geotechnical engineering.

Introduction

Geotechnical engineering construction is affected significantly by varying rock, soil, and underground water conditions and, therefore, must account for many variables that are often complex, vague, and uncertain. As such, geotechnical problem decision making must always be handled by experts' experience. Many techniques, besides experienced estimating, are employed to solve the geotechnical problems. These techniques include expert system, rule induction, case-based reasoning, neural network etc., to inference results based on historical data (Yang 1997; Goh et al. 1995; Jan et al. 2002).

Because human beings are capable of handling uncertain information and of learning from and adapting to their surroundings, they are capable of dealing successfully with highly complicated problems. Also, experience and knowledge accumulation increase an individual's ability to solve problems (Haykin 1999). In geotechnical engineering, engineers accumulate experience in order to prepare to face problems. While an "expert" is regarded as a person of great ability, he typically relies upon his individual perspective and accumulated experience to make decisions. Such decision making, reliant upon experience and knowledge, lacks impersonal criteria and different decisions in the face of the same problem can be made due to the limitations of human memory or for other reasons. Furthermore, an expert's experience and knowledge must be retained and maintained when he or she changes position or leaves the organization (Cheng and Ko 2001; Yau and Yang 1998; Cheng et al. 2002)

Artificial intelligence technology provides a way by which computers can assume responsibility for executing tasks normally requiring human input and control. This technology mimics the way in which the human brain deals with uncertainty and learns from/adapts to the surrounding environment. Artificial intelligence can be used to construct a computer model of how humans make decisions and arrive at solutions to problems. When such models are applied in geotechnical engineering, problems can be addressed and decisions made as they would be by experts. This approach avoids the drawbacks of solo decision making while ensuring the continuance of critical knowledge and experience independent of changes in staff and other human resource factors. Geotechnical decision making improves in terms of both efficiency and accuracy through the aid of artificial intelligence technology (Cheng and Ko 2002, 2003).

Genetic algorithm, fuzzy logic, and neural network are the most popular methods employed in artificial intelligence (Martin and Jain 1999), as these three methods can be applied effectively to many different kinds of problems. Each method presents advantages and drawbacks as well as complements the effectiveness of the others. Recent studies have combined these three methods in a manner that takes advantage of their complementary charac-

[1]Professor, Dept. of Construction Engineering, National Taiwan Univ. of Science and Technology, 4 Keelung Rd., Taipei, Taiwan 10672. E-mail: myc@mail.ntust.edu.tw

[2]Post Doctor, Ecological and Hazard Mitigation Research Center, National Taiwan Univ. of Science and Technology, 4 Keelung Rd., Taipei, Taiwan 10672 (corresponding author). E-mail: tsaihsingchih@gmail.com

[3]Assistant Professor, Dept. of Industrial Engineering and Technology Management, Da-Yeh Univ., Taiwan. E-mail: chko@mail.dyu.edu.tw

[4]Research Assistant, Dept. of Construction Engineering, National Taiwan Univ. of Science and Technology, 4 Keelung Rd., Taipei, Taiwan 10672. E-mail: m8405039@mail.ntust.edu.tw

Note. Discussion open until December 1, 2008. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on November 2, 2006; approved on October 22, 2007. This paper is part of the ***Journal of Computing in Civil Engineering***, Vol. 22, No. 4, July 1, 2008. ©ASCE, ISSN 0887-3801/2008/4-272–280/$25.00.

teristics in order to develop modules that imitate the process of human thinking. Some studies have argued that certain combinations of the three methods can create modules that perform better than any single method alone and that such a comprehensive module is suited to handle real-world problems in a practical and effective manner (Linkens and Nyongesa 1996).

In the comprehensive module, the genetic algorithm is applied either to search for the best interconnection, weights, and topology of neural network or to formulate the membership function (MF) within the fuzzy logic system (Jagielska et al. 1999). Fuzzy logic uses fuzzy sets to describe uncertainties in input and output data. Fuzzy logic is also employed in fuzzy inference (Gorzalczany and Gradzki 2000). The neural network is employed to modify the distribution of membership functions within the fuzzy logic system and also gathers fuzzy rules from cases within the fuzzy system (Ghezelayagh and Lee 1999). However, some of the methods are combined in a model separately. Illegal solutions also occasionally occur in an analysis model. Suboptimal solutions result from a fixed neural network topology being applied to different problems. For these reasons, models remain unable to attain necessary levels of accuracy and suitability when used to solve practical problems (Liska and Melsheimer 1994; Ishigami et al. 1995; Qin et al. 1997).

In this paper, the object-oriented evolutionary fuzzy neural inference system (OO-EFNIS) was developed in order to use artificial intelligence technology to replicate the process of human thinking. In developing the necessary evolutionary fuzzy neural inference model (EFNIM), the genetic algorithm, fuzzy logic, and neural network were combined together. In the module, the genetic algorithm was used within a fuzzy neural network to determine optimal parameters for such things as: membership function distribution, network topology and interconnectivity, connection weights, etc. Fuzzy logic was used to handle approximate inference and uncertainty and the neural network to define the complex relationships between input and output layers. Finally, by applying the object-oriented technique, the EFNIM is transformed by the computer into the OO-EFNIS.

The OO-EFNIS, characterized by intelligent construction and the ability to self-adaptation, was developed to collect engineering data. It imitates the evolution and experiential learning process undergone by geotechnical engineering decision makers and can be applied effectively to assist geotechnical engineers in making correct decisions to practical geotechnical engineering problems. The achievement of this study is its successful application of the proposed OO-EFNIS module to various types of geotechnical engineering decisions.

Brief Introduction to Fuzzy Logic, Neural Network, Genetic Algorithm, and Object-Oriented Technique

Fuzzy Logic

Zadeh (1965) proposed fuzzy logic (FL) to describe uncertainty and imprecision. Because FL imitates the way that the human brain makes decisions by a high order mode when faced with uncertainty or vagueness, it provides an effective way for automated systems to describe highly complex, ill-defined and difficult-to-analyze subjects. In general, fuzzy logic is composed of a fuzzifier, rule base, inference engine, and defuzzifier (Cheng and Ko 2002). In the FL approach, there remain some problems to be overcome such as configuration of distributions of membership functions, determination of composition operator, and acquisition of appropriate fuzzy rules in applications (Maier et al. 2000). Although the FL parameters can be determined through experts' knowledge, a complex problem overcomes difficulties to determine the parameters (Gorzalczany and Gradzki 2000).

Neural Network

With a structure that comprises many neurons, neural network (NN) focuses primarily on computing and storing information. Because NN imitates the human brain in learning, recalling, and generalization, it is usually designed to solve nonlinear and ill-structured problems (Haykin 1999). However, appropriate structures as well as parameters of NN are essential for various kinds of problems. Because the optimal network topology is highly problem oriented, it is difficult to determine (Liatsis and Goulermas 1995). In addition, some real world applications have been hampered by a lack of training techniques which reliably find a global optimum set of weights (Jagielska et al. 1999).

Genetic Algorithm

Genetic algorithm, which imitates parts of the natural evolution process, was first proposed by Holland (1975). GA does not have many mathematical requirements for the optimization problems. In addition, the ergodicity of gene and evolution operations makes GA an effective global search method. Besides, because of its simplistic implementation procedure, the GA provides great flexibility to hybridize domain-dependent heuristics to effectively implement specific problems. For these merits, GA has focused much attention on studying its potential for the optimization technique (Gen and Cheng 1997). However, simple GA is difficult to directly and successfully apply to many difficult-to-solve optimization problems (Michalewicz 1996).

Object-Oriented Approach

"Object" is a physical and conceptual abstraction. Object oriented is a set of objects and interactions between objects. As the OO approach describes objects by abstraction, it is suited for large-scale and complicated systems. OO descriptions are easier to comprehend than detailed descriptions and they view a system as a collection of interacting objects that work together to accomplish tasks (Satzinger et al. 2000). Moreover, objects can be reused within the system and, when objects are changed, the system remains. These advantages, such as extensibility, maintainability, reusability, and configuration management of the specifications, design, analysis, etc. have been instrumental in promoting the OO approach as a major technique in recent system development.

Evolutionary Fuzzy Neural Inference Model

EFNIM Framework

The EFNIM developed in this paper uses the complementary characteristics of genetic algorithm, fuzzy logic, and neural network (see Fig. 1). This combination offsets the demerits of one approach by another's merits. GA is used for optimization; FL deals with uncertainties and handles approximate inferences; and NN is employed in fuzzy input-output mapping.

In this paper, artificial intelligence is used to imitate the decision-making processes in the human brain. EFNIM was developed based on imitating the human high-order inference pro-

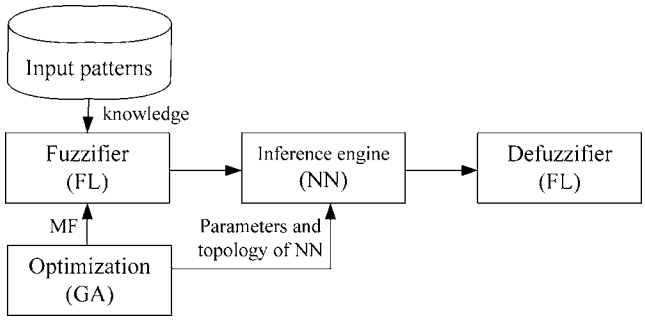

Fig. 1. EFNIM architecture

cess with FL. Although FL can describe the high-order human inference process, making decisions regarding the appropriate distribution of membership functions, operator composition, and regulations is not easy. To resolve this and provide FL with the ability of self-learning, the EFNIM introduces NN. The combination of the FL and NN is regarded as a "neuro with fuzzy input-output," which is also a neural network with both fuzzy inputs and fuzzy outputs. For convenience, the term "neuro with fuzzy input-output" is called fuzzy neural network (FNN) which is a general phrase to express fusion/union of FL and NN (Hayashi et al. 1998). Even if FNN is more applicable than traditional FL in the inference process or single NN in the imitating process, deciding the distribution of MF is difficult and NN parameters such as topology, training rules, etc., cannot be determined easily. GA is an effective approach to conquer the drawbacks of FNN (Gorzalczany and Gradzki 2000). GA, which is applied for optimization in wide territories, addresses the above-mentioned problems by searching for optimal MF and identifying optimum network parameters. With all these constituent components in place, the EFNIM can be constructed.

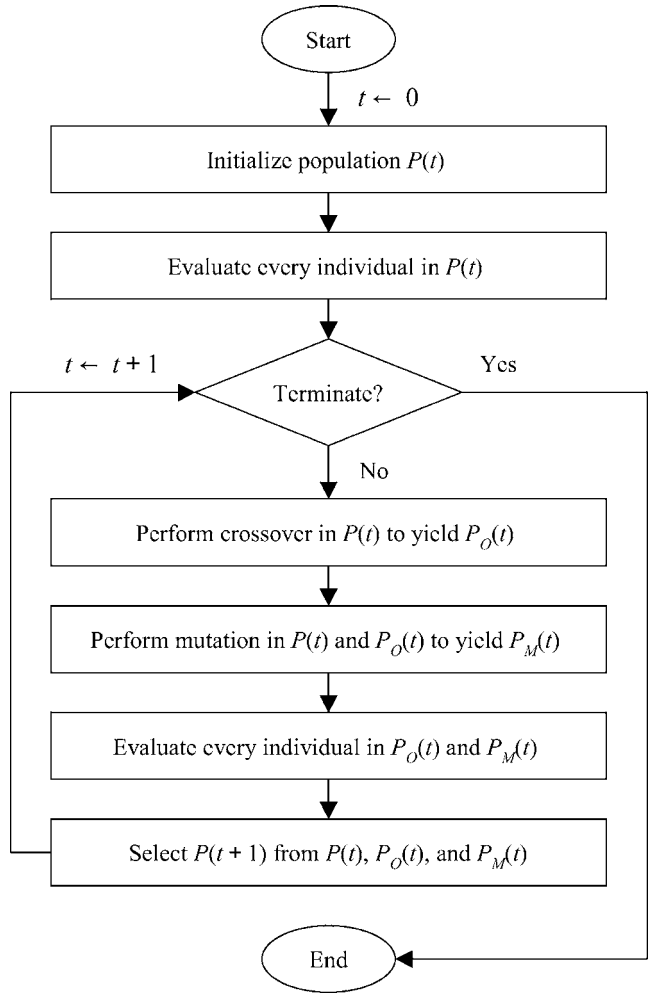

Fig. 2. EFNIM adaptation process

EFNIM Self-Adaptation

The adaptation process of EFNIM is shown in Fig. 2. $P(t)$ is used to represent ξ parents in generation t; $P_0(t)$ means that performing crossover ξ parents yield σ children; $P_M(t)$ means τ mutant individuals.

Initialize Population

The first step of the adaptation process is to generate a set of initial population. In the process, each individual encodes model variables into a binary string to simulate a natural chromosome. Every string comprises two segments: MF substring and NN substring. Two codification methods, summit and width representation method (SWRM) and block representation method (BRM), are employed to encode MF and NN into substrings. The SWRM and BRM encode MF and NN by a fixed and variable substring, respectively. Therefore, the EFNIM encodes the problem using the variable length gene code. The lengths of the substrings depend on the characteristics of the variables including the required variable precision, amount of variables, and variable domains.

Evaluate Individuals

The purpose of evaluation is to evaluate the fitness of chromosomes. At the beginning generation ($t=0$), the EFNIM evaluates ξ individuals. Since the model adopts an enlarged sampling space, the EFNIM evaluates $(\sigma+\tau)$ individuals when $t>0$ where σ is the offspring population and τ is the mutation population.

The aim of the adaptation process is to obtain a model with high accuracy and good generalization properties. The model accuracy on input patterns can be improved by increasing the network complexity. However, an accurate model fit to input patterns does not mean that the overall problem behaviors are captured well. A large network size has higher computational cost. Also, in general, it suffers from overfitting of data in input patterns and deterioration of generalization properties (Maier et al. 2000). Thus, the objective of the EFNIM is to preserve the acceptable prediction accuracy using the fittest shapes of MF with the minimum NN topology and optimum NN parameters, which is posed as an optimization problem. The objective function of the model, f^{ob}, is a combination of model accuracy and model complexity as given in Eq. (1)

$$f^{\mathrm{ob}} = w^{\mathrm{ma}} \times e^{\mathrm{ad}} + w^{\mathrm{mc}} \times \mathrm{mc} \tag{1}$$

$$e^{\mathrm{ad}} = \left(\frac{\sum_{i=1}^{n^{\mathrm{ip}}} \sum_{j=1}^{n^{\mathrm{cn}}} (d_{ij} - a_{ij})^p}{n^{\mathrm{ip}} \times n^{\mathrm{cn}}} \right)^{1/p} \tag{2}$$

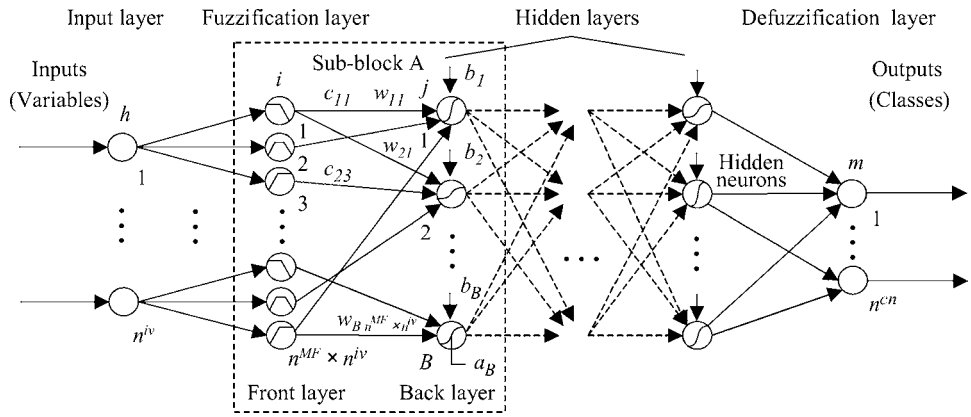

Fig. 3. FNN architecture

$$\mathrm{mc} = \sum^{\text{All sub-blocks}} \sum_{i=1}^{B} \sum_{j=1}^{n^{\mathrm{MF}} \times n^{\mathrm{iv}}} C_{ij} \tag{3}$$

where w^{ma}=related weighting of model accuracy; e^{ad}=prediction error between desired output d_{ij} and actual output a_{ij}; w^{mc} =related weighting of model complexity; and mc=model complexity which is simply formulated by the number of active connections in the network. In Eq. (2), p=power number; n^{ip}=input pattern number; n^{cn}=number of classes in one net; d_{ij}=desired output for the jth class of ith input pattern; a_{ij}=actual output due to the jth class of ith input pattern; and p=power number. In Eq. (3), mc is simply formulated by the number of active connections in the network. n^{MF}=number of fuzzy sets in one complete MF set; and n^{iv}=number of input variables. C_{ij} is 1 or 0 determined by GA to represent the connection exist or not. The meanings of some parameters can be obtained in Fig. 3.

Evaluate Fitness Function

Fitness is a major index to evaluate the status of chromosomes. The bigger fitness value achieves the model objective more. In this study, fitness function is the reciprocal of objective function and is given by Eq. (4)

$$f_k^{\mathrm{ft}} = \frac{1}{f_k^{\mathrm{ob}}} \tag{4}$$

where f_k^{ft} and f_k^{ob}=fitness and objective of kth chromosome.

Perform Crossover

The crossover repeatedly exchanges high performance notations to upgrade performance. It operates on a pair of chromosomes (parents) and produces two children once by exchanging the parent features. The EFNIM uses one-cut-point crossover and exchanges right parts of their parents (see Fig. 4). After crossover, the summit positions of MF, widths of MF, hidden layers, hidden neurons, interconnections, biases, and activation slopes of parents are exchanged.

Perform Mutation

The mutation produces spontaneous random changes in various chromosomes. It protects against premature loss of important notations. For EFNIM, the purpose of mutation is to adjust the value of summits and widths of MF, interconnections, weights, biases, and activation slopes for better performance. It alters one or more genes with a probability (p^{ge}), which is smaller than or equal to mutation rate (p^{mu}). Mutation operation compares the gene's p^{ge} with p^{mu} bit by bit. If $p^{\mathrm{ge}} \leq p^{\mathrm{mu}}$, then the value of the gene will be altered.

Select Individuals

The selection process emulates the survival-of-the-fittest mechanism in nature. It selects a new population with respect to the probability distribution based on fitness for survival. The EFNIM uses a "roulette wheel" method to select fitter chromosomes on an enlarged sampling space. Performing selection on the enlarged sampling space improves GA performance by enlarging the searching space in parallel and by increasing crossover and mutation rates without introducing too much random perturbation. Also, it provides population, children population, and their mutation with the same chance of competing for survival (see Fig. 5).

Evolutionary Fuzzy Neural Inference System

Based on the EFNIM, an object-oriented technique was applied to develop OO-EFNIS. In developing this system, the incremental

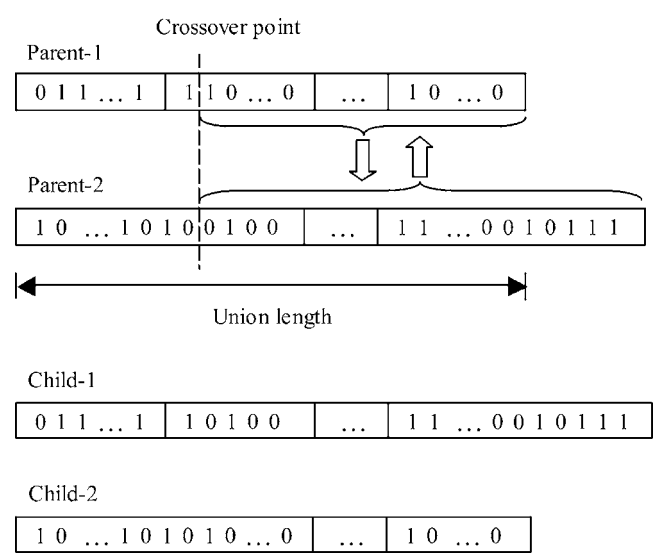

Fig. 4. One-cut-point crossover

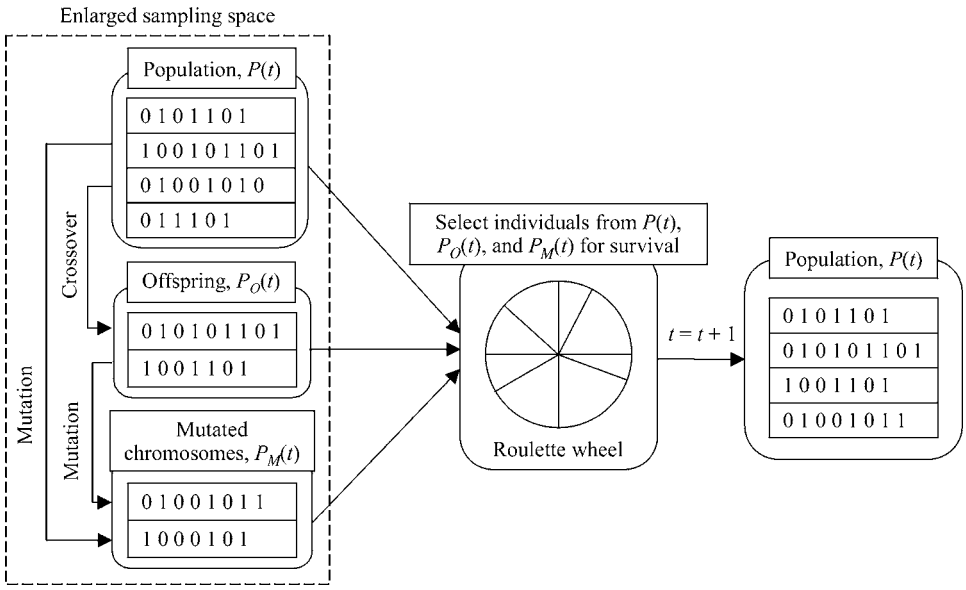

Fig. 5. Selection on enlarged sampling space

and iterative model was adopted and the unified modeling language (UML) was used as the standard software specification language. Analysis and design are the major work of each cycle in development (Larman 1998). The analysis and design of EFNS are illustrated as follows.

Analysis of EFNIS

System analysis makes a description of expecting what work should be done (Satzinger et al. 2000). The emphasis of object-oriented system analysis centers on figuring out the concepts within the problem domain. Therefore, system functions are generalized first in Table 1. Second, a UML package diagram is used to describe system concepts by referencing system functions (see Fig. 6).

Design of EFNIS

System design makes descriptions of expecting how work can be done (Satzinger et al. 2000). In other words, designs serve as guidelines for implementing analysis results. Analysis of an object-oriented system emphasizes the definition of logical software objects. The EFNIS framework is described in Fig. 7. Domain objects serve as the concepts during the analysis process. These concepts are represented by class diagrams in Fig. 8. The design of OO-EFNIS used in this study is described in detail in class diagrams.

Table 1. System Functions

Functions
Handle system parameters.
Handle solutions.
Handle different problems.
Execute the EFNIM adaptation process.
Set up model parameters.
Display model convergence in real time.
Log completed information during model adaptation.
Interrupt the model adaptation process.
Store optimum solutions.
Display information related to the stored solutions.
Compute actual output of any case using any derived solution.
Display actual output and desired output.
Plot the trend of actual output and desired output.

System Validations

The proposed EFNIS in this paper can be applied to assist geotechnical engineers in facilitating decision making when faced with a variety of problems, including slope safety monitoring; predicting deformation in deep excavations; factor identification

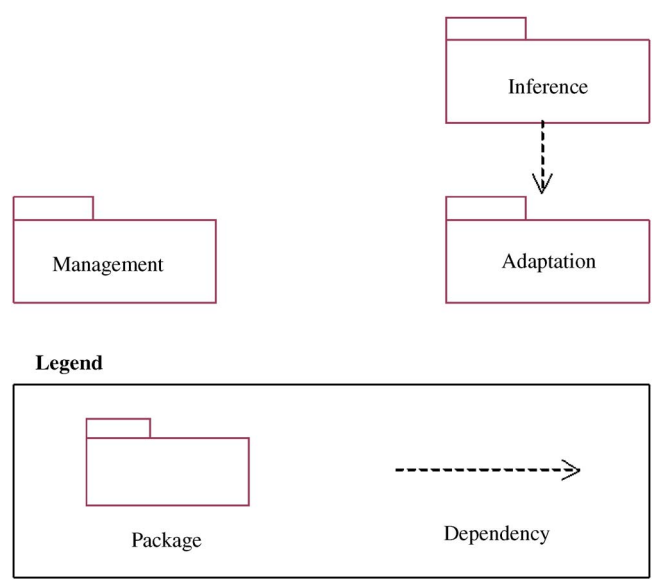

Fig. 6. System concepts

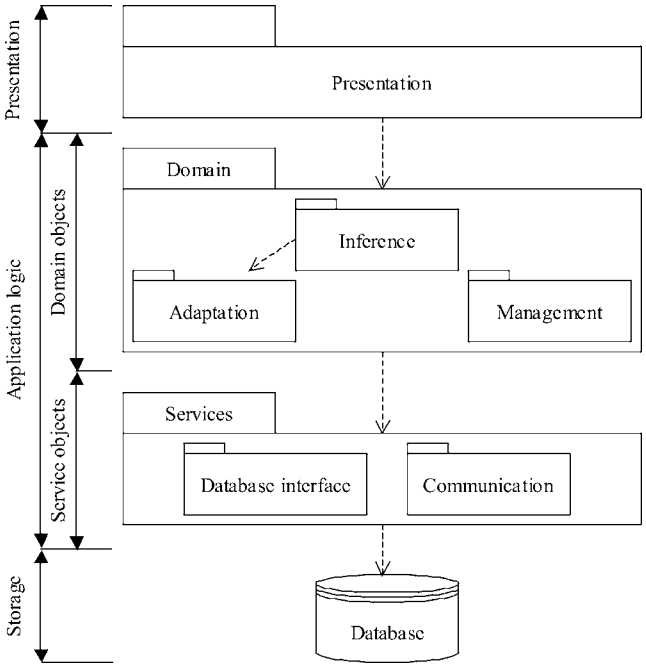

Fig. 7. Multilayered object-oriented system framework

in geotechnical engineering disasters; potential analysis of slope failure; and selecting optimal retaining wall and slope protection methods, among others. The following two example problems show EFNIS' applicability to decision making in geotechnical engineering.

The first case is to estimate the construction duration of a slurry wall. The data were collected by Yang and Yau (2000), in which there were 15 input parameters including site location, excavation depth, wall depth, wall thickness, etc., and the desired output, actual duration, is involved. The estimated construction duration is the only one output in this case study. There were 46 training cases and five testing cases collected. In the reference, three methods: expert system (ES); case-based reasoning (CBR); and intelligent duration estimation system for slurry wall construction (DESW), were applied. Two engines were adhered to the proposed DESW, which were case-based reasoner for case selection and rule-based reasoner for case adjustment.

Selection of retaining wall construction methods is the second case studied herein. There were 254 retaining wall cases collected by Yang (1997). Nineteen features were used to describe the problem, including location, excavation depth, field area, working space, etc. Based on the 19 features, nine inputs can be categorized and ten kinds of construction methods were set as candidates for output, i.e. construction method selection: auger boring pile (ABP); driven pile (DP); full casing pile (FCP); H-section steel pile (HP); open excavation (OE); prepakt mortar pile (PMP); retaining column (RC); steel sheet pile (SSP); steel rail pile (SRP); and slurry wall (SW). Thirty four test cases were collected from different construction firms. In the reference, four methods were applied, including ES; rule induction (RI); CBR; and intelligent retaining wall selection system (RWSS). The RWSS system has three inference engines: a case-based reasoner as a reasoning body; a rule-based reasoner for rule based reasoning; and knowledge-based reasoner with experts' knowledge for the inferring process.

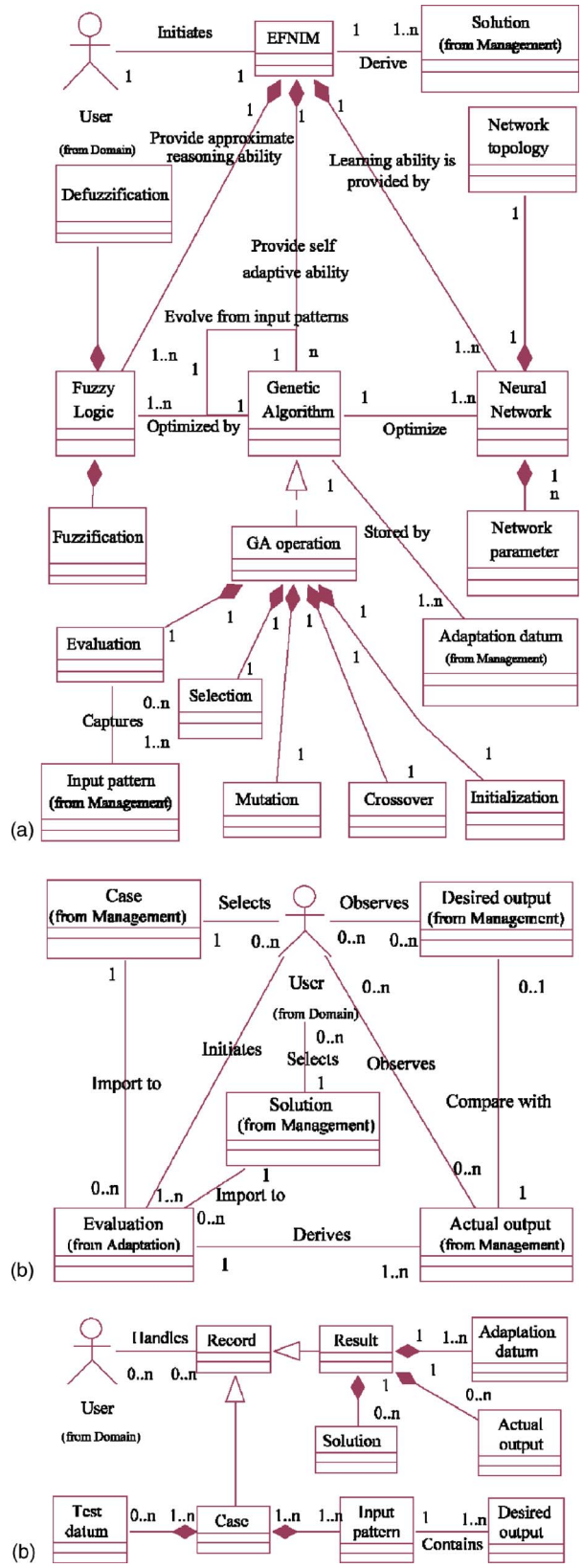

Fig. 8. Class diagrams: (a) adaptation concept; (b) management concept; and (c) inference concept

Slurry-Wall Duration Estimation

Slurry walls are typically an important component of construction projects. The construction duration of a slurry wall, therefore, has

Table 2. Parameters of EFNIS for Slurry-Wall Duration Estimation

Parameters of EFNIS	Values
Input neurons number	15
Output neurons number	1
Grades of input neurons	5
Minimum hidden layers number	1
Maximum hidden layers number	1
Minimum neurons in each layer number	5
Maximum neurons in each layer number	10
Weighting of model accuracy, w^{ma}	1
Weighting of model complexity, w^{mc}	0.0001
Power number, p	2
Type of selected activation function	Logistic sigmoid function
Shape of membership function	Trapezoidal
Total membership function number	15
Crossover rate	0.5
Mutation rate	0.01
Population size	10
Iteration set	5,500

significant bearing on engineering project efficacy and viability. Estimating slurry-wall duration is full of uncertainties and vagueness due to incomplete and complicated data that result from slurry walls being constructed underground. The result is that the duration cannot be easily expressed by a formula and accuracy usually depends on the level of knowledge and experience of the geotechnical engineers doing the estimation. As the EFNIS is a computer system and can replicate the human decision-making process, it is applicable to obtaining reliable slurry-wall duration estimates.

In this case, the data used were gathered from a reported reference case (Yang and Yau 2000). The major parameter settings of EFNIS are listed in Table 2. EFNIS was applied to this case, with a comparison shown in Table 3 of the results of EFNIS analysis and the results obtained through other analytical systems including ES, CBR, and DESW. The result obtained by EFNIS is slightly better than that obtained by DESW and significantly better than achieved by either ES or CBR. This result demonstrates that EFNIS can assist in duration estimation for slurry wall construction in geotechnical engineering decision making.

Selections of Retaining Wall Construction Methods

A temporary/perpetual retaining facility is often made in foundation excavations. A good retaining facility ensures not only the safety and quality during construction but also that proper capability is exhibited after construction is finished. The retaining method selected must consider factors such as safety, constructability, economical feasibility, and environmental impact. Because so many factors play a part in retaining method selection, experts are usually relied upon to make such decisions. The EFNIS helps geotechnical engineers choose an optimal retaining method by learning from past project experience and by mirroring expert decision-making processes.

The parameter settings of EFNIS for this case study are shown in Table 4. In Table 5, the results of ES, RI, CBR, and RWSS from Yang (1997) are compared with the results of EFNIS, where the EFNIS makes perfect percentage of the construction method selection. This comparison proves that the EFNIS can choose a good retaining method for geotechnical engineers.

Table 4. Parameters of EFNIS for Selections of Retaining Wall Construction Methods

Parameters of EFNIS	Values
Input neurons number	9
Output neurons number	1
Grades of input neurons	5
Minimum hidden layers number	1
Maximum hidden layers number	1
Minimum neurons in each layer number	5
Maximum neurons in each layer number	10
Weighting of model accuracy, w^{ma}	1
Weighting of model complexity, w^{mc}	0.0001
Power number, p	2
Type of selected activation function	Logistic sigmoid function
Shape of membership function	Trapezoidal
Total membership function number	15
Crossover rate	0.5
Mutation rate	0.01
Population size	10
Iteration set	5,500

Conclusions

This paper provides a summary introduction to evolutionary fuzzy neural inference system mechanisms, the process of system development, and the experience of application of this system to practical geotechnical engineering problems. The EFNIM comprises three different artificial intelligence technologies, including

Table 3. Test Results of Duration Estimation

		ES		CBR		DESW		EFNIS	
Number of testing cases	Actual duration (day)	Estimated duration (day)	Error (%)	Estimated duration (day)	Error (%)	Estimated duration (day)	Error (%)	Estimated duration (day)	Error (%)
1	96	71	26.0	109	13.5	99	3.1	102.3	6.6
2	72	70	2.8	88	22.2	85	18.1	98.0	36.2
3	127	99	22.0	104	18.1	100	21.3	108.8	14.3
4	83	54	34.9	72	13.3	73	12.0	80.5	3.0
5	80	62	22.5	60	25.0	60	25.0	94.3	17.9
Average error (%)		**21.7**		**18.4**		**15.9**		**15.6**	

Note: Error=|(actual duration-estimated duration)/actual duration| × 100%.

Table 5. Test Results of Construction Method Selections

Testing cases	Actual selection	ES	RI	CBR	RWSS	EFNIS
1	**SW**	RC,ABP, DP	HP,**SW**	**SW**	RC,**SW**,FP	**SW**
2	**HP**	RC,ABP,**HP**	DP,**HP**,SRP	**HP**	RC,DP,**HP**,SRP	**HP**
3	**RC**	**RC**,ABP,DP	**RC**	DP,HP,SRP	**RC**,DP,HP,SRP	**RC**
4	**SW**	**SW**,FCP,RC	**SW**	**SW**	**SW**	**SW**
5	**RC**	**RC**,ABP,FCP	**RC**	**RC**	**RC**,SRP	**RC**
6	**SW**	RC,ABP,**SW**	HP	PMP,**SW**	RC,HP,**SW**,PMP	**SW**
7	**SW**	**SW**,RC,ABP	SRP	**SW**	**SW**,SRP	**SW**
8	**SW**	RC,**SW**,ABP	HP,PMP,**SW**	PMP	HP,PMP,RC,**SW**	**SW**
9	**HP**	RC,ABP,**HP**	**HP**	DP,**HP**	**HP**	**HP**
10	**SW**	RC,ABP,**SW**	**SW**	**SW**	**SW**,RC	**SW**
11	**SW**	ABP,**SW**,RC	DP,HP,**SW**	**SW**	RC,DP,HP,**SW**	**SW**
12	**RC**	**RC**,ABP,DP	—	**RC**	**RC**	**RC**
13	**RC**	**RC**,DP,ABP	HP,SW	SW	SW,HP,HP	**RC**
14	**SRP**	RC,ABP,HP	**SRP**	DP,HP	DP,HP,RC,**SRP**	**SRP**
15	**OE**	RC,SSP,PMP	**OE**	**OE**	**OE**,RC	**OE**
16	**OE**	SSP,ABP,HP	PMP,SW	HP	HP,PMP,SSP,SW	**OE**
17	**SW**	**SW**,FCP,RC	**SW**	**SW**	**SW**	**SW**
18	**SRP**	RC,FCP,SW	**SRP**	**SRP**	**SRP**,RC	**SRP**
19	**SW**	**SW**,FCP,RC	DP,HP,**SW**	**SW**	DP,HP,**SW**	**SW**
20	**SRP**	RC,ABP,SW	**SRP**	DP,HP,**SRP**	RC,DP,HP,**SRP**	**SRP**
21	**SSP**	SW,RC,ABP	SW,**SSP**	SW	**SSP**,SW	**SSP**
22	**SW**	**SW**,FCP,RC	DP,FP,**SW**,SSP	**SW**	DP,HP,SSP,**SW**	**SW**
23	**RC**	**RC**,ABP,FCP	PMP,SRP	HP,DP,SRP	**RC**,HP,PMP,DP,SRP	**RC**
24	**SW**	**SW**,RC,FCP	**SW**	**SW**	**SW**	**SW**
25	**SW**	RC,**SW**,ABP	**SW**	**SW**	RC,**SW**	**SW**
26	**SRP**	RC,ABP,HP	DP,HP	DP,HP,**SRP**	DP,HP,RC,**SRP**	**SRP**
27	**HP**	RC,DP,ABP	DP,**HP**	OE	RC,DP,**HP**,OE	**HP**
28	**SW**	—	DP,HP	SRP	DP,HP,SRP	**SW**
29	**RC**	—	**RC**	**RC**	**RC**	**RC**
30	**HP**	—	—	DP,**HP**	DP,**HP**	**HP**
31	**SW**	—	DP,HP,**SW**	**SW**	DP,HP,**SW**	**SW**
32	**SRP**	—	DP,HP,OE,**SRP**	DP	DP,HP,OE,**SRP**	**SRP**
33	**SRP**	—	DP,HP,PMP,**SRP**,SSP	SW	DP,HP,PMP,**SRP**,SSP,SW	**SRP**
34	**HP**	—	**HP**	DP,**HP**,OE	DP,**HP**,OE	**HP**
Accuracy ratio (%)		**53**	**74**	**68**	**94**	**100**

genetic algorithm for parameter optimization, neural network for learning, and fuzzy logic for imitation of the human decision-making process. During the system developing process, the object-oriented technique, incremental and iterative model, and unified modeling language were used. This study uses genetic algorithm to optimize the ability of the fuzzy neural network to overcome the bottleneck of FNN practice application. This paper further proposes that this advantageous combination of genetic algorithm, fuzzy logic, and neural network be further developed and applied to the field of geotechnical engineering. Finally, two case studies presented demonstrate the applicability of EFNIS to assist decision makers in geotechnical engineering.

Two case studies, i.e., selection of retaining wall construction methods and estimating slurry-wall duration, were studied and comparisons were made followed by a discussion. In the results of slurry-wall duration estimation, EFNIS can estimate more accurately than ES, CBR, and DESW. In the results of selection of retaining wall construction methods, a precise selection of EFNIS reflects its applicability. These comparisons show that EFNIS has a higher capacity for the geotechnical problems than the other AI approaches when comparing the references. In summary, experienced-oriented problems in geotechnical engineering can be solved by EFNIS and historical data to help engineers in making decisions at the project planning and constructing stages.

References

Cheng, M. Y., and Ko, C. H. (2001). "Computer-aided expert decision support for disaster prevention of hillside residents." *J. Chin. Inst. of Civ. Hydraul. Eng.*, 13(1), 39–51.

Cheng, M. Y., and Ko, C. H. (2002). "Computer-aided decision support system for hillside safety monitoring." *Autom. Constr.*, 11(4), 453–466.

Cheng, M. Y., and Ko, C. H. (2003). "Automated safety monitoring and diagnosis system for unstable slopes." *Comput. Aided Civ. Infrastruct. Eng.*, 18(1), 64–77.

Cheng, M. Y., Ko, C. H., and Chang, C. H. (2002). "Computer-aided DSS for safety monitoring of geotechnical construction." *Autom. Constr.*, 11(4), 375–390.

Gen, M., and Cheng, R. (1997). *Genetic algorithms and engineering design*, Wiley, New York.

Ghezelayagh, H., and Lee, K. Y. (1999). "Training neuro-fuzzy boiler identifier with genetic algorithm and error back-propagation." *IEEE Power Engineering Society*, 2, 978–982.

Goh, A. T. C., Wong, K. S., and Broms, B. B. (1995). "Estimation of lateral wall movements in braced excavations using neural networks." *Can. Geotech. J.*, 32(6), 1059–1064.

Gorzalczany, M. B., and Gradzki, P. (2000). "A neuro-fuzzy-genetic classifier for technical applications." *Industrial Technology 2000. Proc., of IEEE Int. Conf.*, 1, 503–508.

Hayashi, I., Umano, M., Maeda, T., Bastian, A., and Jain, L. C. (1998). "Acquisition of fuzzy knowledge by NN and GA." *Knowledge-Based Intelligent Electronic Systems. Proc. KES '98*, Australia, 1, 69–78.

Haykin, S. (1999). *Neural networks: A comprehensive foundation*, Prentice-Hall, New York.

Holland, J. H. (1975). *Adaptation in neural and artificial systems*, The University of Michigan Press, Ann Arbor, Mich.

Ishigami, H., Fukuda, T., Shibata, T., and Arai, F. (1995). "Structure optimization of fuzzy neural network by genetic algorithm." *Fuzzy Sets Syst.*, 71(3), 257–264.

Jagielska, I., Matthews, C., and Whitfort, T. (1999). "An investigation into the application of neural networks, fuzzy logic, genetic algorithms, and rough sets to automated knowledge acquisition for classification problems." *Neurocomputing*, 24(1–3), 37–54.

Jan, J. C., Hung, S. L., Chi, S. Y., and Chen, J. C. (2002). "Neural network forecast model in deep excavation." *J. Comput. Civ. Eng.*, 16(1), 59–65.

Larman, C. (1998). *Applying UML and patterns: An introduction to object-oriented analysis and design*, Prentice-Hall PTR, Upper Saddle River, N.J.

Liatsis, P., and Goulermas, Y. J. P. (1995). "Minimal optimal topologies for invariant higher-order neural architectures using genetic algorithms." *Industrial Electronics, 1995 ISIE '95., Proc., IEEE Int. Symp.*, Greece, 2, 792–797.

Linkens, D. A., and Nyongesa, H. O. (1996). "Learning systems in intelligent control: An appraisal of fuzzy, neural and genetic algorithm control applications." *IEE Proc.: Control Theory Appl.*, 143(4), 367–386.

Liska, J., and Melsheimer, S. S. (1994). "Complete design of fuzzy logic systems using genetic algorithms." *IEEE World Conf. on Computational Intellience, Proc. 3rd IEEE Conf.*, Orlando, 2, 1377–1382.

Maier, H. R., Sayed, T., and Lence, B. J. (2000). "Forecasting cyanobacterial concentrations using B-Spline networks." *J. Comput. Civ. Eng.*, 14(3) 183–189.

Martin, N. M., and Jain, L. C. (1999). "Introduction to neural networks, fuzzy systems, genetic algorithms, and their fusion." *Fusion of neural networks, fuzzy sets, and genetic algorithms: Industrial applications*, CRC, Boca Raton, Fla., 3–12.

Michalewicz, Z. (1996). *Genetic algorithms+data structures=evolution programs*, Springer, Berlin.

Qin, K., Wang, W., and Gong, M. (1997). "A genetic algorithm for the minimum weight triangulation." *Evolutionary Computation, 1997, IEEE Int. Conf.*, Indianapolis, 541–546.

Satzinger, J. W., Jackson, R. B., and Burd, S. D. (2000). "System analysis and design in a changing world." *System analysis and design in a changing world*, Cambridge, Mass.

Yang, J. B. (1997). "An integrated knowledge acquisition and problem solving model for experience-oriented problems in construction management." Ph.D. dissertation, National Central Univ., Taiwan.

Yang, J. B., and Yau, N. J. (2000). "Integrating case-based reasoning and expert system techniques for solving experience-oriented problems." *J. Chin. Inst. Eng.*, 23(1), 83–95.

Yau, N. J., and Yang, J. B. (1998). "Case-based reasoning in construction management." *Comput. Aided Civ. Infrastruct. Eng.*, 13(2), 143–150.

Zadeh, L. A. (1965). "Fuzzy sets." *Information and Control*, 8(3), 338–353.