# Dynamic Prediction of Project Success Using Artificial Intelligence

Chien-Ho Ko, Ph.D.[1]; and Min-Yuan Cheng[2]

**Abstract:** The purpose of construction management is to successfully accomplish projects, which requires a continuous monitoring and control procedure. To dynamically predict project success, this research proposes an evolutionary project success prediction model (EPSPM). The model is developed based on a hybrid approach that fuses genetic algorithms (GAs), fuzzy logic (FL), and neural networks (NNs). In EPSPM, GAs are primarily used for optimization, FL for approximate reasoning, and NNs for input-output mapping. Furthermore, the model integrates the process of continuous assessment of project performance to dynamically select factors that influence project success. The validation results show that the proposed EPSPM, driven by a hybrid artificial intelligence technique, could be used as an intelligent decision support system, for project managers, to control projects in a real time base.

**DOI:** 10.1061/(ASCE)0733-9364(2007)133:4(316)

**CE Database subject headings:** Project management; Predictions; Fuzzy sets; Neural networks; Evolutionary computation; Artificial intelligence.



## Introduction

Success on a project means that certain expectations for a given participant are met, whether owner, planner, engineer, contractor, or operator (Sanvido et al. 1992). The measurements of project success in the construction industry are: Cost, schedule, performance, and safety (Hughes et al. 2004). Hughes et al. (2004) developed a Construction Project Success Survey instrument to identify important success metrics before the start of a project, and to evaluate the level of success achieved at project completion. The measuring metrics include the objective (such as, cost, schedule, performance, and safety) and subjective considerations. Griffith et al. (1999) developed an objective metrics that comprised of four variables: Budget achievement, schedule achievement, design capacity, and plant utilization. The authors discovered that despite the complexities involved in measuring project success, a measurement can be developed based on objective project performance. Shields et al. (2003) established a metric for measuring the success of the construction phase of projects. The metric is quantitatively derived from the Construction Industry Institute's Benchmarking and Metrics (BM and M) database. Scoring ranges from 0–10 based on statistical distributions—for construction cost growth, construction schedule growth, lost workday case incident rates, and rework factor—are presented.

[1]Dept. of Construction Engineering, National Taiwan Univ. of Science and Technology, Taiwan. E-mail: D8805102@mail.ntust.edu.tw

[2]Professor, Dept. of Construction Engineering, National Taiwan Univ. of Science and Technology, Taiwan.

Note. Discussion open until September 1, 2007. 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 October 1, 2004; approved on July 31, 2006. This paper is part of the ***Journal of Construction Engineering and Management***, Vol. 133, No. 4, April 1, 2007. ©ASCE, ISSN 0733-9364/2007/4-316–324/$25.00.

Several studies have also endeavored to reveal factors that influence project success. Lists of variables have abounded in the literature. Chan et al. (2004a, 2005) developed a conceptual framework on critical success factors. Five major groups of independent variables: namely, project-related factors, project procedures, project management actions, human-related factors, and external environment, are identified as crucial to project success. Nguyen et al. (2004) uncovered success factors for large projects, using factor analysis. A survey questionnaire was used to collect data from practitioners. These factors were grouped under four categories: Comfort, competence, commitment, and communication. Through a postal questionnaire survey, Chan et al. (2004b) analyzed a set of success factors using factor analysis and multiple regression. Chua et al. (1997) identified key project management attributes associated with achieving successful budget performance, using a neural network approach. Eight key management factors were identified in the research: (1) number of organizational levels between the project manager and craft workers, (2) amount of detailed design completed at the start of construction, (3) number of control meetings during the construction phase, (4) number of budget updates, (5) implementation of a constructability program, (6) team turnover, (7) amount of money expended on controlling the project, and (8) the project manager's technical experience.

Identification of the success factors could formulate effective strategies for minimizing construction conflicts; whereas, incorporating success factors into developing a research model for predicting the success levels improves project performance. Parfitt and Sanvido (1993) proposed a checklist that can be used by building professionals, as a guideline in predicting the success of a project. Items represented in the checklist include questions similar to the questions and categories initially used to gather information for the identification of the critical project success factors. Chua et al. (1997) developed a predictive model of project success, using neural networks to forecast budget performance of a construction project.

Previous results in measuring and predicting project success show that the measurements of project success are crucial for

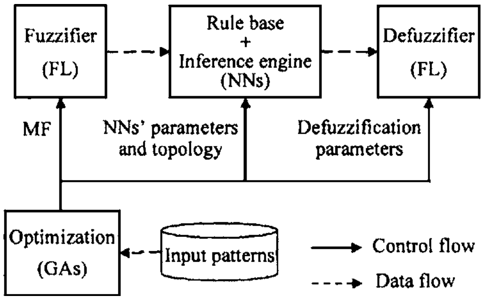


**Fig. 1.** EFNIM architecture

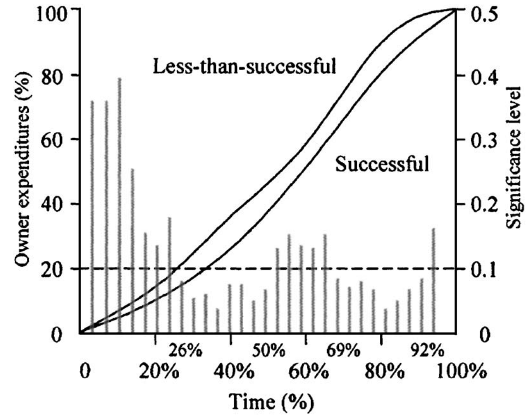


**Fig. 2.** Predictor analysis of owner expenditures (adapted from Russell et al. 1997)

controlling projects, so as to achieve project objectives. Besides, statistical methods provide a theoretical basis for analyzing factors of project success from questionnaire surveys or collected data. However, previous research in predicting project success either adopted fixed factors at various points of time, or used an inference method that has a few difficulties and was inadequate.

From an owner's perspective, the definition of a successful project is one that meets or exceeds budgetary and schedule expectations. On the other hand, a less-than-successful project fails to meet budgetary and/or schedule expectations (CII 1996). Project managers are responsible for project success. In order to ensure that the projects can be accomplished successfully, project managers have to continuously monitor the project performance, so that they can take proper corrective actions to control the project.

Project outcomes are affected by different factors at diverse time points. During the course of a project, predicting project outcomes at different stages requires analysis of dissimilar factors (Russel et al. 1997; Griffith et al. 1999). A dynamic prediction methodology is, thus, required for project managers to continuously monitor project performance. However, every time point has numerous time-dependent variables affecting the project's outcomes. In addition, due to the nature of the construction industry, those variables are uncertain (Barraza et al. 2000). Dynamically predicting the project's outcome under such complex and uncertain circumstances is never easy. Human experts can judge a project's outcome according to their knowledge; however, the significance of these judgments is restricted by their subjective cognitions and/or limited knowledge.

Artificial intelligence (AI) is concerned with building computer systems that solve the problem intelligently by emulating the human brain. AI technology provides techniques for the computer programs to carry out a variety of tasks, at which humans are currently better (Haykin 1999). Consequently, AI paradigms are appropriate for solving project management problems (Ko 2002). The most popular AI paradigms are genetic algorithms (GAs), fuzzy logic (FL), and neural networks (NNs). The combination of GAs, FL, and NNs offsets the demerits of one paradigm by the merits of another (Martin and Jain 1999). In the last few years, several articles have been devoted to the study of fusing GAs, FL, and NNs to derive a better model performance than those using a single conventional method (Linkens and Nyongesa 1996, Cheng and Ko 2003).

## Literature Review

### *Genetic Algorithms*

GAs were first proposed by Holland, in the 1970s, for mimicking some observed natural evolution processes (Holland 1975). The GA is a stochastic searching process based on the mechanism of natural selection and natural genetics. It combines the survival-of-the-fittest principles among strings with a randomized information exchange to form an evolutionary algorithm (Goldberg 1989). Under GA terminology, an individual represents a solution to a problem. The group of individuals at each generation is a population. A generation denotes that a new population of individuals is created. In artificial genetic systems, each individual is represented by a string-named chromosome.

### *Fuzzy Logic*

FL was first developed by Zadeh, in the 1960s, for representing uncertain and imprecise information (Zadeh 1965). In a wide sense, FL is synonymous with fuzzy set theory; that is, the theory of classes with unclear boundaries. In a narrow sense, FL is a logic system that is intended to serve as logic of approximate reasoning (Zadeh 1994). Fuzzy logic systems (FLSs) simulate the high level human decision making process, which aims at modeling the imprecise modes of reasoning to make rational decisions in an environment of uncertainty and imprecision. In general, FLSs contain four major components: Fuzzifier, inference engine, rule base, and defuzzifier (Klir and Yuan 1995).

### *Neural Networks*

The concept of artificial neural networks originated from modeling the brain in the 1950s. Artificial neural networks, commonly referred to as "neural networks," are massively parallel-distributed processors made up of simple processing units (neurons), which perform computations and store knowledge (Haykin 1999). Since the brain is robust, fault tolerant, highly parallel, and able to deal with noisy information, modeling the functions of the brain provides an alternative approach to conventional methods (Aggarwal and Song 1997). NNs model the brain using networks composed of neurons.

### *Evolutionary Fuzzy Neural Inference Model*

One of the pioneering models hybridizing GAs, FL, and NNs for solving construction management problems was developed by Ko (2002). Architecture of the model is shown in Fig. 1. In Fig. 1, the combination of GAs, FL, and NNs offset the demerits of one



J. Constr. Eng. Manage., 2007, 133(4): 316-324

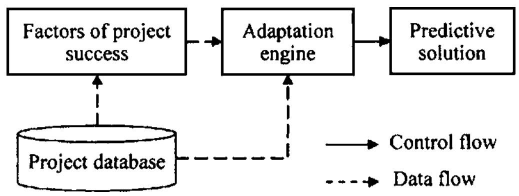


**Fig. 3.** EPSPM architecture

paradigm by the merits of another. FL is primarily concerned with imprecision and approximate reasoning; NN with fuzzy input-output mapping; and GAs with optimization.

The model was developed based on FL that mimics the high level of human inference process. However, FL encounters difficulties in determining fuzzy rules and distributions of membership functions (MFs), while problem complexity increases. NNs are adaptive to changes in the surrounding environment through learning. The nonlinearity of networks captures complex interactions between input and output variables, which provides a promising direction to represent fuzzy rules. The combination, of these two methods into an integrated system, appears to be a promising path toward the development of intelligent systems capable of capturing qualities characterizing the human brain (Canuto et al. 1999; Rajasekaran and Vijayalakshmi Pai 2000). Although, the integration of FL and NN is more reasonable than traditional FL to simulate the characteristics and process of human inference, the NN has demonstrated the difficulty in selecting an appropriate topology for learning different tasks, as well as appropriate parameters, for a network. In addition, the determination of suitable distribution for MFs, for solving disparate problems is time consuming, and the difficulties increase with the problem complexity. To conquer the remaining difficulties, the evolutionary fuzzy neural inference model (EFNIM) employed GA to simultaneously search for the fittest shapes of MFs, optimum NN topology, and optimum parameters of NN, including interconnections status, synaptic weights, bias values, and slopes of activation functions.

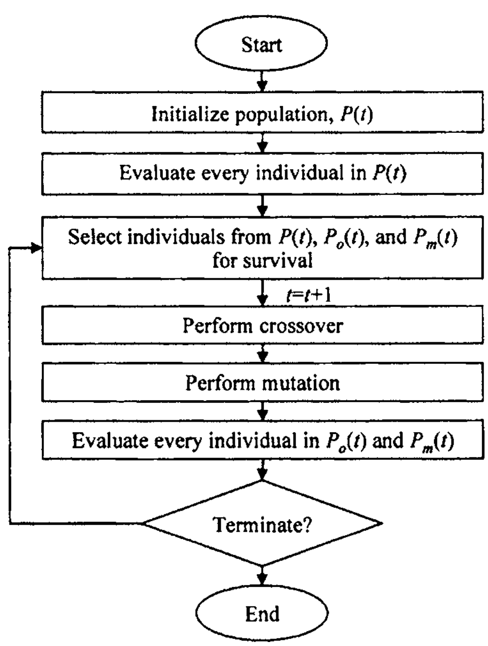


**Fig. 4.** EPSPM adaptation structure

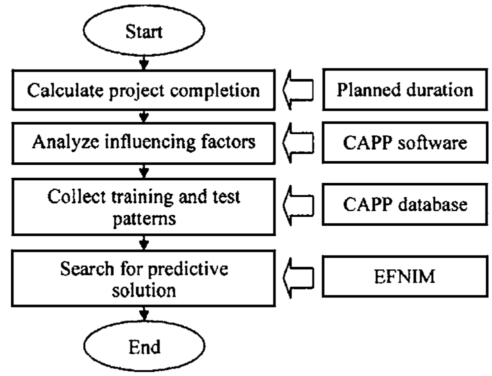


**Fig. 5.** Model application process

## Continuous Assessment of Project Success

To monitor the progress of a project, continuous assessment of project success (CAPP) (Russell et al. 1996) was developed to identify continuous variables that have the ability for predicting project outcome. A continuous variable is a time-dependent quantity whose value can be collected at several points of time during the course of a project (e.g., contractor expenditure, invoice paid by the contractor, owner project commitment, and project cost) (Russell et al. 1996). In the model, a continuous variable is considered as a predictor of project outcome, while its difference between the s-curves for "successful" and "less-than-successful" projects is significant. The significance between the two projects is analyzed through hypothesis tests conducted using the student's $t$ distribution. The significant difference exists, if the value of alpha (Type I error) is less than or equal to 0.10. An alpha equal to 0.10 is frequently used as the critical level for rejecting the null hypothesis. Fig. 2 displays an example analyzing whether or not a variable—owner expenditure—is a predictor at different points in time. In the figure, the $x$-axis represents time from 0 (beginning of design and procumbent) to 100% completion (end of construction) during a project life cycle; while the $y$ axis represents the normalized value of the continuous variable. The vertical bars, denoting the significance level, below 0.10 represents a statistical difference between successful and less-than-successful projects. Observing Fig. 2, variable owner expenditures is a predictor of project outcome from about 26 to 50% and from 69 to 92% project complete.

## Evolutionary Project Success Prediction Model

### *Model Architecture*

Project success can be predicted at different points in time with their corresponding factors while it is progressing. In the beginning of a project, the procurement of material and equipment effects the outcome of the project. By contrast, in the construction phase, project success can be predicted by the factor of owner expenditures. Previous projects provide patterns for analyzing those factors and project success. The architecture of EPSPM shown in Fig. 3 contains four components: The project database, factors of project success, adaptation engine, and predictive solution. The project database contains previous cases of projects that serve as patterns for analysis. The element of the factors of a project's success determines which factors are effective for predicting the project outcome at a specific point in time. Com-

**Table 1.** Time-Dependant Variables Analyzed by CAPP

| Variable name | Sign. level |
|---|---|
| Actual design % complete | 0.03 |
| Actual owner expenditures | 0.00 |
| Cost of contractor project commitments | 0.01 |
| Cost of owner project commitments | 0.04 |
| Actual owner effort hours | 0.03 |
| Recordable incident rate (by period) | 0.01 |
| Actual overtime work | 0.02 |
| Cost of change orders | 0.08 |
| Quantity of change orders | 0.02 |
| Days lost to weather (gross working days) | 0.08 |

Note: Significance level below 0.10 represents a statistical difference between successful and less-than-successful. Variable names are defined in Russel et al. (1996).

plex relationships between the factors and the project outcome are automatically identified through previous patterns, using an adaptation engine. Finally, the derived solution can be used as a predictive model for forecasting a project's outcome in a real-time base.

### Model Adaptation Process

The adaptation engine is driven by EFNIM through GAs. The adaptation process is shown in Fig. 4. In the process, $P(t)$ denotes a population at generation $t$, $P_o(t)$ is an offspring population at generation $t$, and $P_m(t)$ indicates a mutation population at generation $t$. Each procedure is defined in the next sections.

### Initialize Population

The first step of the adaptation process is to randomly generate a set of initial solutions. Each solution encodes model variables into a binary string to simulate a natural chromosome. Every string is comprised of two segments: MF substring and NN substring. Two codification methods—summit and width representation method (SWRM) and block representation method (BRM)—proposed by Ko (2002) are employed to encode MFs and NNs into substrings.

### Evaluate Individuals

The aim of the adaptation process is to obtain a predictive model with high accuracy and good generalization properties. The model accuracy on input patterns, acquired from the project's database, 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 an overfitting of the data in the input patterns and deterioration of generalization properties (Maier et al. 2000). Thus, the objective of the adaptation engine is to preserve the acceptable prediction accuracy—using the fittest shapes of MFs with the minimum NN topology and optimum NN parameters, which is posed as an optimization problem. The objective function of the model, $f^{\text{ob}}$, is a combination of the model accuracy and model complexity, as given in Eq. (1)

$$f^{\text{ob}} = c^{\text{aw}} \times s^{\text{er}} + c^{\text{cw}} \times mc \tag{1}$$

where $c^{\text{aw}}$=accuracy weighting coefficient; $s^{\text{er}}$=prediction error between actual output and desired output; $c^{\text{cw}}$=complexity weighting coefficient; and mc is the model complexity, which is simply formulated by the number of active connections in the network.

**Table 2.** Patterns for Dynamic Prediction of Project Success (Shown and Reused CAPP Database with CII's Kind Permission)

| Pattern number | Input patterns: Input (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) | Output (11) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.0000 | 0.9700 | 0.5322 | 0.9253 | 0.0000 | 0.7555 | 0.0000 | 0.0000 | 0.7877 | 0.7719 | 1.0000 |
| 2 | 0.0000 | 1.0000 | 0.5625 | 0.0000 | 0.9372 | 0.6954 | 0.3103 | 0.0000 | 0.0000 | 0.0000 | 1.0000 |
| 3 | 0.3125 | 1.0000 | 0.0000 | 0.7581 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 1.0000 | 1.0000 | 0.0000 |
| 4 | 0.0000 | 0.9950 | 0.0000 | 0.9555 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.8665 | 0.6304 | 1.0000 |
| 5 | 0.0000 | 0.0000 | 0.4694 | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.6667 |
| 6 | 0.0000 | 0.0000 | 0.0000 | 0.7098 | 0.0000 | 0.0000 | 0.0000 | 0.3102 | 0.0000 | 0.0000 | 0.0000 |
| 7 | 0.5897 | 0.9767 | 0.0000 | 0.0000 | 0.9598 | 0.7708 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 1.0000 |
| 8 | 0.0000 | 0.9592 | 0.0000 | 0.8629 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 1.0000 |
| 9 | 0.0000 | 0.9150 | 0.8931 | 0.9170 | 0.9960 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.6667 |
| 10 | 0.0000 | 1.0000 | 0.3808 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 1.0000 |
| 11 | 0.7647 | 0.0000 | 0.4137 | 0.0000 | 0.5522 | 0.5869 | 0.0000 | 0.1507 | 0.0000 | 0.0000 | 1.0000 |
| 12 | 0.0000 | 0.0000 | 0.8654 | 0.0000 | 0.9813 | 0.7909 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.3333 |
| Test patterns | | | | | | | | | | | |
| 13 | 0.0000 | 1.0000 | 0.0000 | 0.7472 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | 0.9612 | 0.9348 | 0.0000 |
| 14 | 0.0000 | 1.0000 | 0.0000 | 0.9537 | 1.0000 | 0.0000 | 0.0000 | 0.1448 | 0.0000 | 0.0000 | 0.6667 |
| 15 | 0.0000 | 0.0000 | 0.0000 | 0.8574 | 0.8396 | 0.0000 | 0.0000 | 0.0000 | 0.5000 | 0.5000 | 1.0000 |

Note: The captions of the above columns are: (1) actual design % complete; (2) actual owner expenditures; (3) cost of contractor project commitments; (4) cost of owner project commitments; (5) actual owner effort hours; (6) recordable incident rate; (7) actual overtime work; (8) cost of change orders; (9) quantity of change orders; (10) day lost to weather; and (11) project performance.



J. Constr. Eng. Manage., 2007, 133(4): 316-324

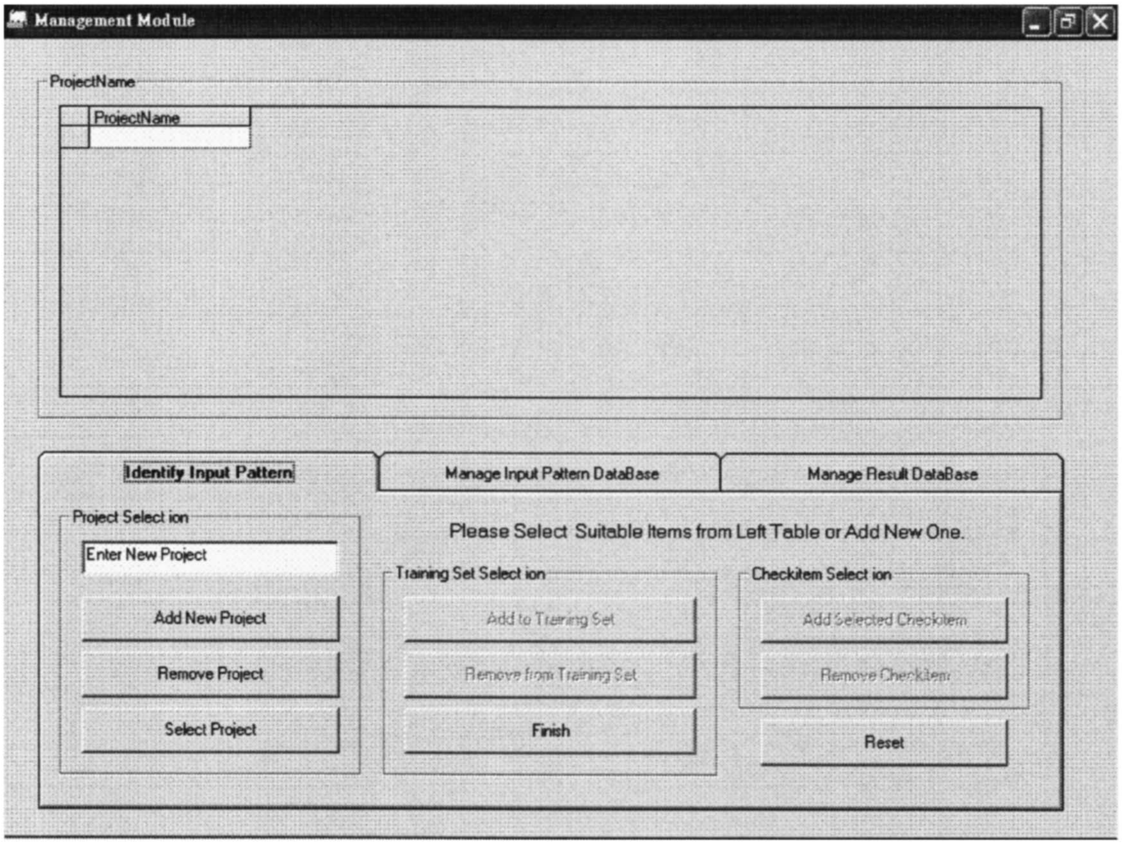


**Fig. 6.** EPSPM management module

### *Evaluate Fitness Function*

Fitness is a major index to evaluate the status of the chromosomes. A larger fitness value indicates the result more closely achieves the model objective. In this research, fitness function is the reciprocal of objective function and is given by Eq. (2)

$$v^{\mathrm{fi}}{}_k = \frac{1}{v^{\mathrm{ob}}{}_k} \tag{2}$$

where $v^{\mathrm{fi}}{}_k$ and $v^{\mathrm{ob}}{}_k$=fitness and objective values of chromosome $k$.

### *Perform Crossover*

The crossover repeatedly exchanges high performance notations in the search for better and better performances. The adaptation engine uses one-cut-point crossover and exchanges the right parts of their parents. After crossover, the summit positions of MFs, widths of MFs, hidden layers, hidden neurons, interconnections, biases, and activation slopes of parents are exchanged.

### *Perform Mutation*

The purpose of mutation is to adjust the value of summits and widths of MFs, 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 $p^{ge} \leq p^{mu}$, 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. Since genetic operators are blind in nature, to avoid fitter chromosomes from being lost in the evolutionary process, the adaptation engine uses a modified enlarged sampling space that contains all of the population, offspring population, and their mutation.

### *Model Application Process*

Specific processes and methods used to implement EPSPM are summarized in Fig. 5. Referring to Fig. 5, the blocks on the left-hand side are the procedures used to implement the model. The blocks on the right-hand side are the detailed methods and attributes concerned with the execution of the tasks on the left-hand side.

#### Calculate Project Completion

To analyze the factors of success in the next step, progress is converted to a project completion. Time is normalized such that 0 percent and 100 percent can correspond to a specific date. A ratio of to-date to planned is used in this investigation. For example, a ratio of 100 days passed the date of the beginning to a duration of 1000 days is 10% completion.

#### Analyze Influencing Factors

The factors of success, at a specific completion, are analyzed using CAPP software. Hypothesis tests with student's t distribu-



J. Constr. Eng. Manage., 2007, 133(4): 316-324

tion are used to identify influencing factors, whose value of alpha (type I error) less than or equal to 0.10 are selected. In statistical research, an alpha equal to 0.1 is frequently used to reject the null hypothesis. CAPP software analyzes 76 continuous variables based on 54 construction projects.

### Collect Training and Test Patterns

Training and test patterns are collected from CAPP database that contains 54 construction projects. These projects are real data collected by Russell et al. (1996) from the 16 representative Construction Industry Institute (CII) member companies.

### Search for Predictive Solution

A predictive model (i.e., fuzzy neural network model) is adopted using EFNIM according to the identified factors of success with a 54 projects database. The model is specifically generated to predict project outcomes at current project progress (project complete). More details about the application of EFNIM can be found in Ko and Cheng (2003).

## *Model Validation*

### *Model Implementation*

Performance of the proposed EPSPM is validated using a 67% completion project. The 67% completion is targeted to coincide with those examples previously explained in the CAPP related literatures (Russell et al. (1996); Russell et al. (1997); CII (1996)). Using the CAPP software, 10 of 76 time-dependent variables whose significance level below 0.1 at 67% completion were identified, as summarized in Table 1.

The original patterns are acquired from CAPP system database with CII's kind permission. The database contains 54 projects, which are the same data used to analyze the significance level. This study selects 15 of 54 available projects from the database where construction type is process plant projects, designer contractor type is lump sum construction contract, and the stability of projects is at the completed stage. In AI field, 80% available patterns for training purpose, and 20% for test purpose are frequently used. The 12 of them (80%) are, thus, treated as input patterns for evolvement, and the other three projects (20%) are selected as test patterns. Since this validation attempts to predict the project outcomes at 67% completion, values of time-dependent variables at 67% completion are collected. The normalized data of 15 projects are shown in Table 2. In the table, four categories of project performance defined by CAPP software, namely successful, on time, or on budget, less-than-successful, and disastrous are linearly represented using 0, 0.3333, 0.6667, and 1, respectively.

EPSPM provides three modules for application: Adaptation module, inference module, and management module. The 15 selected projects, with 10 identified factors of success, are entered into EPSPM through management module (see Fig. 6). Adaptation module shown in Fig. 7 is then used to evolve a predictive model for 67% project completion. To preserve the acceptable prediction accuracy without suffering overfitting of data, a policy that reduces model complexity without deteriorating prediction