JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT
ISSN 1392-3730 print/ISSN 1822-3605 online
2014 Volume 20(2): 223–236
doi:10.3846/13923730.2013.801891

INTERVAL ESTIMATION OF CONSTRUCTION COST AT COMPLETION USING LEAST SQUARES SUPPORT VECTOR MACHINE

Min-Yuan CHENG[a], Nhat-Duc HOANG[b]

[a]Department of Construction Engineering, National Taiwan University of Science and Technology, Taiwan
[b]Faculty of Building and Industrial Construction, National University of Civil Engineering, Hanoi, Vietnam

Received 28 Oct 2011; accepted 08 Jun 2012

Abstract. Completing a project within the planned budget is the bottom-line of construction companies. To achieve this goal, periodic cost estimation is vitally important not only in the planning phase, but also in the execution phase. Due to high uncertainty in operational environment, point estimation of project cost is oftentimes not sufficient to assist the decision-making process. This study utilizes Least Squares Support Vector Machine (LS-SVM), machine learning based interval estimation (MLIE), and Differential Evolution (DE) to establish a novel model for predicting construction project cost. LS-SVM is a supervised learning technique used for regression analysis. MLIE is employed for inference of prediction intervals. Moreover, our model deploys DE in the cross validation process to search for the optimal values of tuning parameters. The newly developed model, named as EAC-LSPIM, yields results consisting of a point estimate coupled with lower and upper prediction limits, at a certain level of confidence, to accentuate uncertainty. Simulation and performance comparison demonstrate that the new model is capable of delivering accurate and reliable forecasting results.

Keywords: construction management, prediction interval, estimate at completion, least squares support vector machine, differential evolution, machine learning.

Reference to this paper should be made as follows: Cheng, M.-Y.; Hoang, N.-D. 2014. Interval estimation of construction cost at completion using least squares support vector machine, *Journal of Civil Engineering and Management* 20(2): 223–236. http://dx.doi.org/10.3846/13923730.2013.801891

Introduction

In construction industry, project success has always foundered on the high uncertainty in operational environment. Thus, it is not surprising that construction projects frequently suffer cost overrun (Nassar *et al.* 2005). In order to operate profitably, construction companies must frequently evaluate project cost at completion to detect deviations and to carry out appropriate responses. However, construction firms typically focus on budget planning during the initial project stage, which practically ignores the impact of engineering cost changes and information updates during construction (Cheng *et al.* 2010). This fact prevents effective project cost control and detection of potential problems. Therefore, cost estimation is a crucial task and it needs to be carried out at various stages of a project (Liu, Zhu 2007). Moreover, the accuracy of construction cost estimation is a critical factor in the success of the project (Kim *et al.* 2004). Poor cost estimation may result in profit loss and occasionally leads to project failure.

Due to its importance, various predictive methods have been proposed for cost estimation. Approaches that are applicable to cost estimation range from statistics based multivariable regression analysis to machine-learning techniques such as Classification and Regression Trees (CART), M5 model tree (M5-MT), Artificial Neural Network (ANN), Support Vector Machines (SVM), and Least Squares Support Vector Machine (LS-SVM).

Multivariable regression analysis is a very powerful statistical tool that can be used as both an analytical and a predictive technique in assessing the contribution of potential new items to the overall estimation, although it is limited in modeling non-linear relationships (Kim *et al.* 2004). In addition, when the number of input variables becomes considerably large, the prediction performance of this method often deteriorates significantly.

CART (Breiman *et al.* 1984) is a classification method which utilizes historical data to construct decision trees. A CART model that forecasts the value of continuous variables from a set of input variables is known as a regression-type model (Razi, Athappilly 2005). One major advantage of the decision tree based model is its ability to handle small-size data set. Moreover, CART can mitigate the negative effect of outliers because the model is capable of isolating the outliers in a separate node. However, one disadvantage of CART is that it may produce unstable decision trees (Timofeev 2004). The reason is that insignificant modification of learning sample could result in radical changes in the decision tree. In addition, previous works (Razi, Athappilly 2005) have indicated that prediction performance of CART can be inferior to ANN.

Corresponding author: Nhat-Duc Hoang
E-mail: M9705805@mail.ntust.edu.tw, ducxd85@yahoo.com

A model tree (MT) is similar to a decision tree, but includes the multivariate linear regression functions at the leaves and is able to predict continuous numeric value attributes (Shrestha, Solomatine 2006; Witten, Frank 2000; Kaluzny *et al.* 2011). The algorithm separates the parameter space into subspaces and constructs a local linear regression model in each of them. Thus, MT is, to some degree, similar to a piecewise linear function. In the M5-MT, the nodes of the tree are selected over the attribute that maximizes the expected error reduction as a function of the standard deviation of output parameter (Bonakdar, Etemad-Shahidi 2011). MT is proved to have the capability of learning in an efficient manner and it can tackle regression tasks with high dimensionality. Compared to other machine learning techniques, MT training process is relatively fast and the results are interpretable (Shrestha, Solomatine 2006).

ANN is a viable alternative for forecasting construction costs and in practice, it has been used to construct various cost prediction models (Hegazy, Ayed 1998; Zhu *et al.* 2010; Sonmez 2011). This method eliminates the need to find a mapping relationship that mathematically describes the construction cost as a function of input variables. When the influence factors and the structure of ANN are all specified, the task boils down to collecting a reasonable number of data to train the ANN. However, the training process of ANN based models is often time-consuming; and ANN also suffers from difficulties in selecting a large number of controlling parameters such as hidden layer size, learning rate, and momentum term (Bao *et al.* 2005).

Furthermore, one major disadvantage of ANN is that its training process is achieved through a gradient descent algorithm on the error space, which can be very complex and may contain many local minima (Kiranyaz *et al.* 2009). Thus, the training of ANN is likely to be trapped into a local minimum and this definitely hinders the forecasting capability. To overcome such issue, evolutionary algorithms, such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), can be used to train the ANN model (Nasseri *et al.* 2008; Zhang *et al.* 2007). It is because these advanced optimization techniques can significantly reduce the chance of getting trapped in local minima. Hence, the training process possibly settles in an optimum solution; nevertheless, this cannot be guaranteed (Kiranyaz *et al.* 2009).

In construction area, SVM has been utilized in cost estimation (Cheng *et al.* 2010; Kong *et al.* 2008; An *et al.* 2007; Hongwei 2009). The principles of SVM are based on the structural risk minimization and statistical learning theory. The SVM based models also involve identification of influence factors, collection of data sample, and training/testing process. After the mapping function has been established, the model is capable of predicting the future value of project cost. The advantages of SVM are widely known including strong inference capacity, excellent generalization, and accurate prediction ability (Lam *et al.* 2009; Huang *et al.* 2004). Nevertheless, SVM training process entails solving a quadratic programming problem subjected to inequality constraint. This means that SVM's training process for large data sets requires expensive computational cost (Guo, Bai 2009).

To overcome the drawback of SVM, LS-SVM has been proposed recently by Suykens *et al.* (2002), Gestel *et al.* (2004), and Brabanter *et al.* (2010). LS-SVM is a modified version of SVMs to alleviate the burden of computational cost. In LS-SVM's training process, a least squares cost function is proposed to obtain a linear set of equations in the dual space. Consequently, to derive the solution, it is required to solve a set of linear equations, instead of the quadratic programming as in standard SVM. And, this linear system can be efficiently solved by iterative methods such as conjugate gradient (Wang, Hu 2005). Studies have been carried out to demonstrate the excellent generalization, prediction accuracy, and fast computation of LS-SVM (Yu *et al.* 2009; Samui, Kothari 2011; Chen *et al.* 2005). Despite of its superiority, application of LS-SVM in construction cost estimation is still very limited.

Additionally, when applying LS-SVM, it is recognizable that the tuning parameters, namely regularization and kernel function parameters, play an important role in establishing the predictive model (Yu *et al.* 2009; Suykens *et al.* 2002). These parameters control the model's complexity, and they are needed to be determined properly via cross-validation. In doing so, the main objective is to obtain an optimal model that can explore the underlying input-output mapping function and is capable of producing the best predictive performance on new data (Bishop 2006). In this study, DE, a population-based stochastic search engine proposed by Storn and Price (Price *et al.* 2005), is employed in the cross-validation process to achieve such objective.

In practice, cost estimation in construction industry is often stated in the form of a point forecast (Trost, Oberlender 2003; Iranmanesh *et al.* 2007; Cheng *et al.* 2010; Zhu *et al.* 2010). However, decision makers require not only accurate forecasting of certain variables but also the uncertainty associated with the forecasts. Point estimation does not take into account the various sources of uncertainty that stem from the model itself, input variables, and tuning parameters. Thus, incorporating prediction uncertainty into deterministic forecasts can help improve the reliability and the credibility of the model outputs (Shrestha, Solomatine 2006).

Various approaches (Wonnacott, T. H., Wonnacott, R. J. 1996; Heskes 1997; Mencar *et al.* 2005; Brabanter *et al.* 2011) have been introduced to achieve interval estimation. However, existing methods also have many limitations such as requiring the prior distributions of the uncertain input parameters or data and involving certain assumptions about the data and error distribution. The accuracy and the credibility of those approaches rely significantly on the precision of prior information and their assumptions. Another class of methods for deriving prediction interval (PI) is relied on re-sampling or bootstrap. Although bootstrap based methods (Sonmez 2011; Stine 1985; Lam, Veall 2002) can yield accuracy prediction result, this method is notably characterized by high computational cost.

Recently, a new framework for estimation of PI which is based on machine learning technique has been established by Shrestha and Solomatine (2006). The proposed method does not require any assumption and prior knowledge of input data or model error distribution. Moreover, it also does not demand intensive computational cost as in bootstrap based methods. In their research work (Shrestha, Solomatine 2006), the superiority of machine-learning based interval estimation (MLIE) over traditional methods is exhibited.

Therefore, this study aims to propose an artificial intelligence model, namely as EAC-LSPIM, that hybridizes various advanced techniques including LS-SVM, MLIE, and DE to help project manager in construction cost prediction. The newly built model incorporates the strengths and mitigates the weaknesses of each individual technique. The research goal is to build a model that can operate automatically without human intervention and can deliver accurate and reliable forecasting results. Equipped with this tool, it is expected that the tasks of cost control and cost planning in construction industry can be carried out effectively.

The remaining part of this paper is organized as follows. The second section of this paper reviews related research works on estimating of construction cost at completion, LS-SVM, techniques for achieving prediction intervals, and DE. In the third section, the DE-based cross-validation process is introduced. The fourth section describes the framework of the newly proposed model in detail. Simulation and result comparison of the model are demonstrated in the fifth section.

1. Review of pertinent literature

1.1. Estimate of project cost at completion

In construction management, estimating cost of work at completion is vitally important for project success. To achieve this, project managers often rely on Earned Value Management (EVM) methodology. EVM is widely known as a management technique that relates resource planning and schedule usage and technical performance requirement (Abba 1997). EVM comprises of three essential components that support project control: Plan Value (PV) or Budgeted Cost of Work Schedule (BCWS), Earned Value (EV) or Budgeted Cost of Work Performed (BCWP), and Actual Cost (AC) or Actual Cost of Work Performed (ACWP). In the construction industry, project managers emphasize the application of EVM as it provides a tool for tracking project status and for measuring project performance.

EVM is a systematic approach to forecast Estimate at Completion (EAC). The role of EAC is accentuated due to the fact that managers or planners can appraise the total project cost based on the estimated value of EAC. Iranmanesh *et al.* (2007) point out that the correct and the on time EAC is essential for preventive response during the project execution. If EAC indicates an overrun in cost, the project managers can use proper strategies to adjust construction cost. In the situation of cost overrun, project managers arguably carry out a value engineering program for cost reduction in which scope or quality in some sections of project is decreased. Another option is to require additional budget to offset overrun cost.

At every completion period, managers can extract data from progress report, calculate project Earned Value (EV) and predict EAC. The EAC can be computed by formula using cost management data provided by the contractor in the Cost Performance Report or the Cost/Schedule Status Report. The reliability of these reports depends on the degree to which the contractor adheres to internal controls involving measuring performance on a contract (Christensen 1993).

According to previous works by Christensen (Christensen 1993; Christensen *et al.* 1995) and Chen (2008), determining an appropriate estimation of EAC is an arduous task. To obtain EAC, managers need to collect voluminous cost management data provided by the contractor in progress report, usually monthly report. For the contractors, in order to form the periodic report to the owner, their site-engineers must gather sufficient data summarized in the daily man-hours summary, daily material summary and daily equipment summary. Finally, one can use various formulas to compute EAC based on the combination of several data elements presented in the report: BCWS, BCWP, and ACWP.

To forecast the EAC, numerous index-based formulas have been utilized. Those formulas are divided into three categories: non-performance method, performance method, composite method (Christensen *et al.* 1995; Chen 2008; Cheng *et al.* 2010). Based on a survey carried out by Christensen *et al.* (1995), the accuracy of index-based formulas depends significantly on the type of system, and the stage and phase of project. This interprets why performance of a particular formula might be quiet acceptable in a certain case, while it could be much worse in other cases (Cheng *et al.* 2010). Project planners must employ their own judgments to ascertain a most appropriate EAC or a range of reasonable EACs. Currently, there is no official guidance on how to choose an amenable EAC calculation according to a specific setting.

Besides index-based formulas, other EAC prediction methods are based on regression analysis (Iranmanesh *et al.* 2007; Christensen *et al.* 1995). The regression-based formulas are typically derived using linear or nonlinear univariate regression analysis (Christensen 1993). However, methods based on traditional regression analysis also have disadvantages such as their limitations in describing nonlinear relationships (An *et al.* 2007). In addition, the number of influence factors for construction cost estimate can be appreciable (Trost, Oberlender 2003; Cheng *et al.* 2010) and the underlying regression function is possibly very intricate. That fact explains why EAC estimation based on traditional regression analysis is not widely used in the industry (Christensen 1993).

Needless to say, EAC prediction problem is complicated since it involves voluminous construction data, considerable number of influence factors, and complicated regression function. Thus, it is reasonable for planners or managers to resort to more advanced methods, specifically Artificial Intelligence (AI) methods, such as Artificial Neural Network (ANN) and Least Squares Support Vector Machine (LS-SVM).

1.2. Least squares support vector machine for regression analysis

This section is dedicated to describing the LS-SVM's mathematical formulation. Consider the following model of interest, which underlies the functional relationship between a response variable and one or more independent variables (Suykens *et al.* 2002; Wang, Hu 2005):

$$y(x) = w^T \phi(x) + b, \tag{1}$$

where: $x \in R^n$, $y \in R$, and $\phi(x) : R^n \to R^{nh}$ is the mapping to the high dimensional feature space. In LS-SVM for regression analysis, given a training dataset $\{x_k, y_k\}_{k=1}^N$, the optimization problem is formulated as follows:

$$\text{minimize} \quad J_p(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2 ; \tag{2}$$

subjected to $y_k = w^T \phi(x_k) + b + e_k$, $k = 1, ..., N$,

where: $e_k \in R$ are error variables; $\gamma > 0$ denotes a regularization constant.

In the above optimization problem, it is noted that the objective function includes a sum of squared fitting error and a regularization term. This cost function is similar to standard procedure in training feedforward neural networks and is related to a ridge regression (Wang, Hu 2005). However, when *w* becomes infinite dimensional, one cannot solve this primal problem. Therefore, it is necessary to construct the Lagrangian and derive the dual problem (Suykens *et al.* 2002).

The Lagrangian is given by:

$$L(w,b,e;\alpha) = J_p(w,e) - \sum_{k=1}^{N} \alpha_k \{w^T \phi(x_k) + b + e_k - y_k\}, \tag{3}$$

where: α_k are Lagrange multipliers. The conditions for optimality are given by:

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \to w = \displaystyle\sum_{k=1}^{N} \alpha_k \phi(x_k) \\ \dfrac{\partial L}{\partial b} = 0 \to \displaystyle\sum_{k=1}^{N} \alpha_k = 0 \\ \dfrac{\partial L}{\partial e_k} = 0 \to \alpha_k = \gamma e_k, \quad k = 1, ..., N \\ \dfrac{\partial L}{\partial \alpha_k} = 0 \to w^T \phi(x_k) + b + e_k - y_k = 0, \quad k = 1, ... N \end{cases} . \tag{4}$$

After elimination of *e* and *w,* the following linear system is obtained:

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \omega + I/\gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{5}$$

where: $y = y_1, ..., y_N$, $1_v = [1;...;1]$, and $\alpha = [\alpha_1;...;\alpha_N]$. And the kernel function is applied as follows:

$$\omega = \phi(x_k)^T \phi(x_l) = K(x_k, x_l). \tag{6}$$

The resulting LS-SVM model for function estimation is expressed as:

$$y(x) = \sum_{k=1}^{N} \alpha_k K(x_k, x_l) + b, \tag{7}$$

where: α_k and *b* are the solution to the linear system (5). The kernel function that is often utilized is Radial Basis Function (RBF) kernel. Description of RBF kernel is given as follows:

$$K(x_k, x_l) = \exp(\frac{\|x_k - x_l\|^2}{2\sigma^2}), \tag{8}$$

where: σ is the kernel function parameter.

In the case of the Radial Basis Function kernel, there are two tuning parameters (γ, σ) that are needed to be determined in LS-SVM. The regularization parameter (γ) controls the penalty imposed to data points that deviate from the regression function. Meanwhile, the kernel parameter (σ) affects the smoothness of the regression function. It is worth noticing that proper setting of these tuning parameters is required to ensure desirable performance of the prediction model (Suykens *et al.* 2002).

1.3. Regression analysis with prediction intervals

1.3.1. Background

Regression analysis is the study of the function that underlies the relation between the dependent variable *Y* and a vector *x* as the independent variable (Olive 2007). A typical regression model can be expressed as follows:

$$Y_i = m(x_i) + e_i, \quad i = 1, ..., n, \tag{9}$$

where: *m* denotes a function of *x* and e_i is the prediction error.

Various methods are used to find the estimate $\hat{m}$ of *m*. These methods range from traditional techniques, such as multiple linear regression model and many time series, nonlinear, nonparametric, and semiparametric models (Olive 2007), to various machine learning techniques, such as M5-MT (Bhattacharya, Solomatine 2005; Jekabsons 2010), ANN (Zhu *et al.* 2010; Wong *et al.* 1997), SVM (Cheng *et al.* 2010; Lu *et al.* 2009), and LS-SVM (Suykens *et al.* 2002; Brabanter *et al.* 2010).

Once the mapping function is obtained, the primary task is to predict the future value of *Y* when a specific input *x* is presented to the system. In point estimation, *Y* is expressed as a single value. On the contrary, in interval estimation, the prediction result is given in the form of an interval of possible values. In many situations, interval estimation draws more attention than point estimation. The reason is that the requirement of decision makers not only resides in an accurate forecasting but also in the inherent uncertainty of the forecasts (Shrestha, Solomatine 2006).

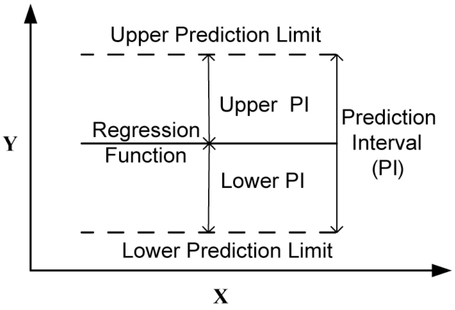

Fig. 1. Prediction limits and prediction interval

Interval estimation includes the upper and lower limits between which a pointwise value of response variable is expected to lie with a certain level of confidence (usually 95%). The range restricted by those limits is known as prediction interval (PI) (Fig. 1). Prediction intervals as outputs are desirable since they provide a range of values that most likely include the point estimation of the predicted variable. In addition, one can employ prediction intervals to discern the accuracy of the estimation provided by the model, and then decide to keep or reject the result (Mencar *et al.* 2005).

1.3.2. Evaluating performance of prediction interval

Once the output with interval has been obtained, the Prediction Interval Coverage Probability (PICP) can be utilized for performance evaluation (Shrestha, Solomatine 2006; Khosravi *et al.* 2010). PICP measures the proportion of data point lying within the PI. In some cases, the empirical PICP can be much less than the pre-specified level of confidence. This phenomenon indicates that the derived PIs are not reliable (Khosravi *et al.* 2011). Hence, PICP is oftentimes expected to be equal or greater than the level of confidence, since this reflects the reliability of the prediction results.

However, PICP is not the only metric for evaluating PIs. The reason is that one can simply construct a very large PI to achieve the maximum reliability of the prediction outcomes (e.g. 100%). Nevertheless, extremely large PIs, in practice, reduce the usability of forecasting results because the interval estimation does not convey any valuable information for the decision-makers (Khosravi *et al.* 2011). Hence, to guarantee the usability of the interval estimation, Mean Width of Prediction Interval (MPI) (Khosravi *et al.* 2010; Shrestha, Solomatine 2006), which measures the average width of the PIs, is also needed to be considered. Accordingly, a well-constructed PI should achieve the balance between reliability and usability. Put differently, it is desirable to obtain a high PICP corresponding to a narrow MPI (Khosravi *et al.* 2010, 2011). Nevertheless, these two criteria oftentimes conflict with each other and this makes interval estimation a challenging problem. Due to its importance and challenge, studies have dedicated in establishing PIs for variety of prediction models.

2.3.3. Previous works on prediction interval estimation

Sonmez (2011) integrated neural networks with bootstrap prediction intervals for range estimation of construction costs. In this approach, neural networks are used for modeling the mapping function between the influence factors and costs. Bootstrap method is utilized to quantify the level of variability included in the estimated costs. However, to construct interval estimates based on the bootstrap, which possibly produces accurate intervals, requires heavy computational expense (Brabanter *et al.* 2011).

Mencar *et al.* (2005) proposed a method for estimating prediction interval for neuro-fuzzy network such that the system provides an estimate of the uncertainty associated with predicted output values. This method does not require any strict assumption on the unknown distribution of data. However, the derived intervals are constant throughout the input domain. This feature might not reflect the true phenomenon happening in real-world time series data. In these cases, inherent uncertainty may distribute unequally in different periods of time (Cheng, Roy 2011).

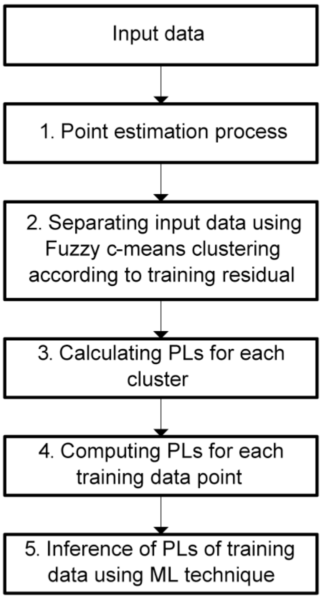
Input data
1. Point estimation process
2. Separating input data using Fuzzy c-means clustering according to training residual
3. Calculating PLs for each cluster
4. Computing PLs for each training data point
5. Inference of PLs of training data using ML technique

Fig. 2. Machine learning (ML) based interval estimation

Another method for constructing PI, which is based on machine learning approach, is established by Shrestha and Solomatine (2006). In their study, the authors presented a method to estimate PI via uncertainty of the model output. The crucial idea herein is the historical residuals between the model outputs and the corresponding observed data can be the quantitative indicators of the difference between the model and the modeled real world system and provide the valuable information to evaluate the model uncertainty.

The machine learning based interval estimation (MLIE) approach (Shrestha, Solomatine 2006) can be divided into five main steps (Fig. 2). First, the point estimation process is carried out. A regression technique is employed to learn the underlying mapping function between input data and outputs. Second, the input data points are separated into different clusters that have similar historical residuals, which are obtained from point estimation process, using fuzzy c-means clustering. In the third step, prediction limits (PLs) for each cluster are computed based on empirical distributions of the errors associated with all data points of one cluster. In the next step, PLs for each training data point is then calculated according to their membership grades in each cluster. In the final step, a machine learning (ML) technique can be deployed to learn the underlying functions between the input data and the computed PLs for training data. PLs for testing data can be inferred using those underlying functions.

Another advantage of MLIE method is its independence on the machine learning technique. However, this approach lacks a mechanism for selecting tuning parameters of the regression machine appropriately. Additionally, the performance of the proposed MLIE, in term of prediction accuracy and of computational cost, can be enhanced significantly by using more superior technique such as LS-SVM.

1.4. Differential Evolution optimization algorithm

This section describes the standard algorithm of Differential Evolution (DE) proposed by Storn and Price (Price *et al.* 2005; Storn, Price 1997). The algorithm (Fig. 3) consists of five main stages: initialization, mutation, crossover, selection, and stopping condition verification. Given that the problem at hand is to minimize a cost function *f(X)*, where the number of decision variables is *D*, we can describe each stages of DE in details.

1.4.1. Initialization

DE commences the search process by randomly generating *NP* number of D-dimensional parameter vectors $X_{i,g}$ where $i = 1, 2, \ldots, NP$ and *g* denotes the current generation. In original DE algorithm, *NP* does not change during the optimization process (Storn, Price 1997). Moreover, the initial population (at $g = 0$) ought to cover the entire search space in a uniform manner. Thus, we can simply generate these individuals as follows:

$$X_{i,0} = LB + rand[0,1] \times (UB - LB), \quad (10)$$

where: $X_{i,0}$ is the decision variable *i* at the first generation. *rand*[0,1] denotes a uniformly distributed random number between 0 and 1. *LB* and *UB* are two vectors of lower bound and upper bound for any decision variable.

1.4.2. Mutation

A vector in the current population (or parent) is called a target vector. Hereafter, the terms parent and target vector are used interchangeably. For each target vector, a mutant vector is created according to the following equation (Storn, Price 1997):

$$V_{i,g+1} = X_{r1,g} + F(X_{r2,g} - X_{r3,g}), \quad (11)$$

where: *r*1, *r*2, and *r*3 are three random indexes lying between 1 and *NP*. These three randomly chosen integers are also selected to be different from the index *i* of the target vector. *F* denotes the mutation scale factor, which controls the amplification of the differential variation between $X_{r2,g}$ and $X_{r3,g}$. $V_{i,g+1}$ represents the newly created mutant vector.

1.4.3. Crossover

The crossover stage aims to diversify the current population by exchanging components of target vector and mutant vector. In this stage, a new vector, named as trial vector, is created. The trial vector is also called the offspring. The trial vector can be formed as follows:

$$U_{j,i,g+1} = \begin{cases} V_{j,i,g+1}, & if \ rand_j \le Cr \ or \ j = rnb(i) \\ X_{j,i,g}, & if \ rand_j > Cr \ and \ j \ne rnb(i) \end{cases}, \quad (12)$$

where: $U_{j,i,g+1}$ is the trial vector; *j* denotes the index of element for any vector; $rand_j$ is a uniform random number lying between 0 and 1; *Cr* is the crossover probability, which is needed to be determined by the user; *rnb(i)* is a randomly chosen index of {1, 2, ..., *NP*} which guarantees that at least one parameter from the mutant vector ($V_{j,i,g+1}$) is copied to the trial vector ($U_{j,i,g+1}$).

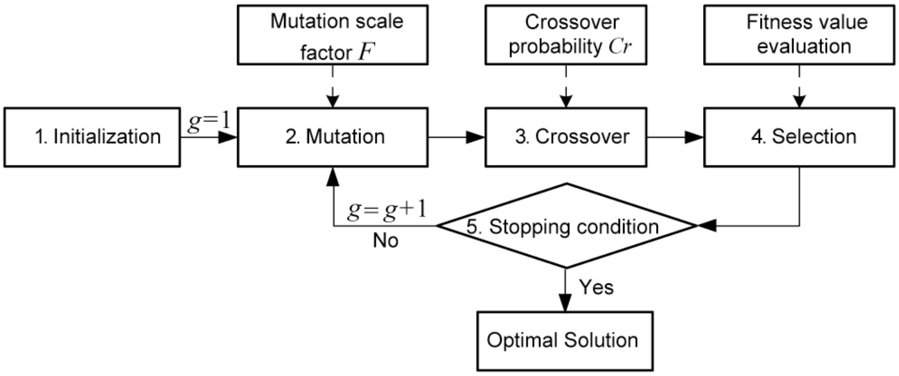

Fig. 3. Differential Evolution optimization algorithm

1.4.4. Selection

In this stage, the trial vector is compared to the target vector (Price *et al.* 2005). If the trial vector can yield a lower objective function value than its parent, then the trial vector replaces the position of the target vector. The selection operator is expressed as follows:

$$X_{i,g+1} = \begin{cases} U_{i,g} \ \ if \ \ f(U_{i,g}) \le f(X_{i,g}) \\ X_{i,g} \ \ if \ \ f(U_{i,g}) > f(X_{i,g}) \end{cases}. \quad (13)$$

1.4.5. Stopping criterion verification

The optimization process can terminate when the stopping criterion is met. The user can set the type of this condition. Commonly, maximum generation (G_{max}) or maximum number of function evaluations (*NFE*) can be used as the stopping condition. When the optimization process terminates, the final optimal solution is readily presented to the user.

2. Differential evolution based cross-validation

As mentioned earlier, in machine learning, one important objective is to construct a prediction model that can deliver the best generalization. The reason is that the performance on the training data set is not necessarily a good indicator of predictive performance on the testing data due to the problem of over-fitting (Bishop 2006). Over-fitting arises when a regression model fits the training set very well, but performs poorly on the new data set.

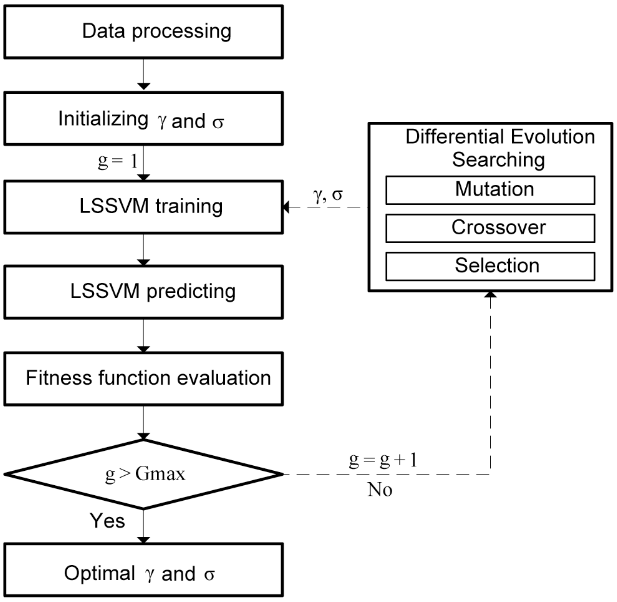

Fig. 4. Differential Evolution based cross-validation process

Hence, to build a desirable prediction model, one commonly used technique is S-fold cross-validation (Bishop 2006; Samarasinghe 2006; Suykens *et al.* 2002). The training data is divided into S folds and this allows a proportion $(S-1)/S$ of the available data to be used for training while other portion of the data is for assessing model performance. However, one major disadvantage of cross-validation is that the number of training runs that must be performed is increased by a factor of *S*, and this can impose difficulty for models with high computational expense in the training process (Bishop 2006). Moreover, another challenge is that there might be infinite combinations of model's parameters. Thus, it is problematic and time-consuming when designing the combinations of parameters for the cross-validation process.

Since our study employs LS-SVM as the regression machine, there are two parameters needed to be determined, namely regularization parameter γ and RBF kernel parameter σ. To avoid over-fitting and drawbacks of traditional cross-validation approach, the new model utilizes DE (Price *et al.* 2005) to automatically explore the various combinations of (γ, σ) and to identify the optimal set of these tuning parameters. In the following section, the DE-based cross-validation (Fig. 4) is described in details.

In the step of data processing, the training data set is divided into *S* folds (e.g. 5 folds). In each run, one fold is used as a validating set; meanwhile, the other folds are used for training the model (Fig. 5). Tuning parameters of LS-SVM is initialized randomly using Eqn (10). The lower bounds for γ and σ are both 0.001. Meanwhile, the upper bounds for γ and σ are specified to be 10000 and 100, respectively.

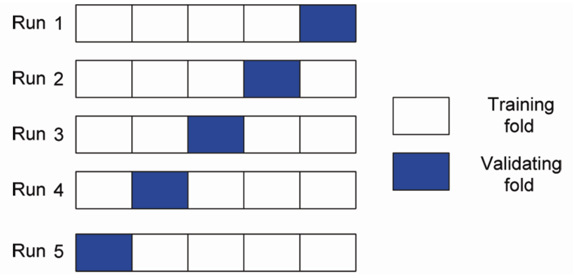

Fig. 5. S-fold cross-validation

In LS-SVM training, LS-SVM is utilized to learn the regression function between input and output for each run. These regression functions can be described in the form of Eqn (7). After the training process, LS-SVM is applied to predict the output of the validating sets. In order to determine the optimal set of tuning parameters, the following objective function is used in the step of fitness function evaluation:

$$F_{fitness} = \frac{\sum_{k=1}^{5} E_{tr}^{k}}{5} + \frac{\sum_{k=1}^{5} E_{va}^{k}}{5}, \quad (14)$$

where: E_{tr}^{k} and E_{va}^{k} denotes the training error and validating error, respectively, for k^{th} run. The training and validating errors herein are Root Mean Squared Error calculated as follows:

$$RMSE = \sqrt{\sum_{j=1}^{N} \frac{(Y_P^j - Y_A^j)^2}{N}}, \quad (15)$$

where: Y_P^j and Y_A^j denote predicted and actual value for output j^{th}. In addition, N is the number of training data in each run.

Table 1. EAC prediction's influencing factors

No.	Influence Factor (IF)	Index	Definition
IF1	Construction duration	Construction progress (%)	Duration to date/ revised contract duration
IF2	Actual cost	AC_p	Actual Cost/ Budget at Completion
IF3	Planned cost	EV_p	Earned Value/ Budget at Completion
IF4	Cost management	CPI	Earned Value/Actual Cost
IF5	Time management	SPI	Earned Value/Planned Value
IF6	Subcontractor management	Subcontractor billed index	Subcontractor billed amount/ Actual Cost
IF7	Contract payment	Owner billed index	Owner billed amount/ Earned Value
IF8	Change order	Change order index	Revised contract amount/ Budget at Completion
IF9	Construction price fluctuation	CCI	Construction material price index of that month/ construction material price index of initial stage
IF10	Number of rainy day	Climate effect index	(Revised project duration – number of rainy day)/ revised project duration

The fitness function, in essence, represents the trade-off between model generalization and model complexity. It is worth noticing that well-fitting of the training set may reflect the model complexity. However, complex model tends to suffer from over-fitting (Bishop 2006; Suykens *et al.* 2002). Thus, incorporating the error of the validating data can help identify the model that features the balance of minimizing training error and generalization property.

In each generation, the DE optimization carries out mutation, crossover, and selection process to guide the initial population to the final optimal solution. The search terminates when the current generation g achieves the maximum number of generation G_{max}. After being optimized, the prediction model is ready to be used in the next step.

3. Interval estimation of construction cost at completion using LS-SVM inference model (EAC-LSPIM)

Figure 6 provides the overall picture of the model EAC-LSPIM. Before describing the model in detail, it is noted that our study benefits from previous research works of Chen (2008) and Cheng *et al.* (2010) in identifying the influence factors for EAC prediction (Table 1), and of Shrestha and Solomatine (2006) in establishing the MLIE.

3.1. Input data

Historical data sets (Table 2) used in this paper were collected from 13 reinforced concrete building projects executed between 2000 and 2007 by one construction company headquartered in Taipei City, Taiwan. Building heights ranges from 9 to 17 stories (including underground floors). Contract values ran from NT$80 million to NT$1.1 billion. Total floor areas for the projects ranged from 2,094 m^2 to 31,797 m^2. Besides, construction durations varied between 15 to 63 months. Historical data sets were separated into training sets (from 1 to 11) and testing sets (12 and 13). The training and testing data sets consist of 262 and 44 data cases respectively. Table 3 provides descriptive statistics of influencing factors as well as desired output of the historical data. In Table 4, the sample of 10 input variables from project 2, which had 24 completion periods, is used to illustrate the data set.

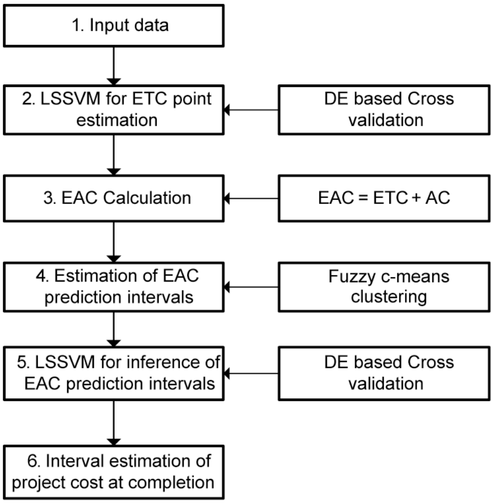
1. Input data
2. LSSVM for ETC point estimation
DE based Cross validation
3. EAC Calculation
EAC = ETC + AC
4. Estimation of EAC prediction intervals
Fuzzy c-means clustering
5. LSSVM for inference of EAC prediction intervals
DE based Cross validation
6. Interval estimation of project cost at completion

Fig. 6. EAC-LSPIM

3.2. LS-SVM for point estimation of Estimate to Completion

Herein, LS-SVM is employed to learn the mapping function between model's input and output. Each 1×10 vector of influence factor acts as input for LS-SVM. Input vectors and observed values of Estimate to Completion (ETC) take the role of training data to obtain the prediction model. LS-SVM uses regularization parameter (γ) and RBF kernel parameters (σ), which are chosen by DE-based cross-validation process. After the training process, the model is capable of inferring unknown ETC value whenever new input information is presented.

Table 2. Project information

Project	Total area (m^2)	Under-ground floors	Ground floors	Buildings	Start date	Finish date	Duration (days)	Contract amount (NTD)	Prediction periods
1	12622	2	9	1	2003/12/1	2005/8/22	630	289,992,000	29
2	4919	3	11	1	2003/12/13	2005/11/10	689	149,300,000	24
3	19205	5	8	1	2000/5/20	2002/5/19	729	332,800,000	20
4	5358	3	9	1	2000/11/15	2002/11/14	729	199,600,000	25
5	27468	2	11	3	1999/12/16	2001/12/3	718	1,142,148,388	26
6	31797	2	9	4	2001/7/4	2003/3/31	635	530,000,000	20
7	7707	2	14	1	2001/11/24	2003/10/20	695	153,500,000	22
8	10087	3	14	1	2002/6/18	2004/7/6	749	216,000,000	27
9	3479	1	10	1	2003/6/2	2004/9/30	486	85,714,286	18
10	7289	2	8	1	2005/6/15	2006/9/15	457	190,844,707	20
11	6352	4	11	1	2004/3/5	2006/2/18	715	202,241,810	31
12	4774	2	11	1	2004/2/21	2006/2/20	730	145,377,589	27
13	3094	2	7	1	2005/10/1	2007/2/28	515	102,500,000	17

Table 3. Descriptive statistics of historical data

	IF1(%)	IF2(%)	IF3(%)	IF4	IF5	IF6	IF7	IF8	IF9	IF10	ETC(%)	EAC(%)
Mean	65.48	51.62	60.66	1.20	1.00	1.06	0.89	1.03	1.05	0.89	42.93	93.87
Median	65.30	49.25	58.05	1.16	1.00	1.08	0.91	1.00	1.03	0.90	42.00	91.70
Minimum	2.40	0.00	0.00	0.34	0.40	0.00	0.00	0.87	0.97	0.70	0.00	73.20
Maximum	130.60	132.70	141.60	2.33	1.62	1.90	1.57	1.42	1.20	1.00	108.70	132.70
Std. Dev.	32.92	31.50	35.72	0.23	0.09	0.26	0.22	0.10	0.06	0.07	30.81	14.84

Table 4. Input data for project 2 with 24 completion periods

IF1	IF2	IF3	IF4	IF5	IF6	IF7	IF8	IF9	IF10
2.4	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6.9	2.5	0.0	1.0	1.0	1.2	1.0	1.0	1.0	1.0
11.0	6.3	11.0	1.8	1.0	1.8	1.0	1.0	1.1	1.0
15.5	9.6	12.7	1.3	1.0	1.3	1.0	1.0	1.1	1.0
19.9	12.6	14.8	1.2	1.0	1.0	0.9	1.0	1.1	1.0
24.2	14.3	16.4	1.1	1.0	0.9	0.8	1.0	1.1	1.0
28.7	16.5	16.4	1.0	1.0	1.4	1.4	1.0	1.1	0.9
33.0	19.4	18.8	1.0	1.0	1.2	1.2	1.0	1.1	0.9
37.4	22.3	26.5	1.2	1.0	1.4	1.2	1.0	1.1	0.9
41.8	25.3	30.4	1.2	1.0	1.4	1.1	1.0	1.1	0.9
46.1	29.2	29.7	1.0	1.0	1.3	1.3	1.0	1.1	0.9
50.6	32.6	33.1	1.0	1.0	1.2	1.2	1.0	1.1	0.9
54.9	36.4	37.0	1.0	1.0	1.3	1.2	1.0	1.1	0.9
59.3	40.6	41.5	1.0	1.0	1.1	1.1	1.0	1.1	0.9
63.5	43.8	44.8	1.0	1.0	1.0	1.0	1.0	1.1	0.9
72.2	54.6	55.8	1.0	1.0	1.1	1.1	1.0	1.1	0.9
76.5	61.5	62.9	1.0	1.0	1.1	1.0	1.0	1.1	0.8
81.0	66.1	67.6	1.0	0.9	1.0	1.0	1.0	1.1	0.8
85.2	72.5	82.5	1.1	1.0	1.1	1.0	1.0	1.1	0.8
89.7	78.5	92.0	1.2	1.0	1.1	1.0	1.0	1.1	0.8
94.1	79.6	94.4	1.2	1.0	1.2	1.0	1.0	1.1	0.8
98.4	81.6	99.0	1.2	1.0	1.2	1.0	1.0	1.1	0.8
102.9	84.5	100.5	1.2	1.0	1.2	1.0	1.0	1.1	0.8
107.2	91.2	100.5	1.1	1.0	1.1	1.0	1.0	1.1	0.8

3.3. Estimate at Completion calculation

In this step, Actual cost percentage (AC) of completed jobs is added to the Estimate to Completion (ETC) in order to obtain the Estimate at Completion (EAC) values, as defined in Eqn (16):

$$EAC_p = AC_p + ETC_p, \tag{16}$$

where: EAC_P denotes point estimation of EAC; ETC_P represents point estimation of ETC; AC_P is actual cost percentage.

ETC is a value used to determine forecasted expenditures necessary to complete remaining project work. AC percentage is a known value defined as the ratio of actual construction cost (AC) value to the Budget at Completion (BAC). It is noted that the BAC itself is the cost of the project when all contracted works are completed. EAC replaces BAC for the predicted total cost of the project at a specific period during construction.

3.4. Estimation of EAC prediction interval

Prediction interval (PI) estimation is carried out in four steps. First, the input dataset is separated into a certain number of clusters corresponding to distributions of historical residuals, obtained from the point estimation, using fuzzy c-means clustering algorithm (FCMC) (Bezdek 1981). The FCMC is an unsupervised machine learning technique employed to separate data into different clusters. Notably, using this technique, a data point can belong to many clusters, and the degrees of belonging are quantified by fuzzy membership grades. In FCMC, the number of clusters needs to be specified by the user. Commonly, the optimal number of clusters can be selected so that it results in the clustering performance corresponding to the smallest Xie-Beni index. For details of FCMC and selecting cluster number, readers are guided to the previous works of Xie and Beni (1991), and of Oliveira and Pedrycz (2007).

The next step is to compute the lower and upper PIs for each cluster. Given a certain level of confidence (e.g. 95% or α is 5%), the PIs for each cluster is calculated from empirical distributions of the corresponding historical residuals (e). To construct $(100-\alpha)\%$ PI, the $(\alpha/2)\times 100$ and $(1-(\alpha/2))\times 100$ percentile values are taken from empirical distribution of residuals for lower and upper PI, respectively (Fig. 7). The mathematical expression for calculating lower and upper PIs for cluster i (PI_{ci}^{L} and PI_{ci}^{U}) is given as follows:

$$PI_{ci}^{L} = e_j \quad j : \sum_{k=1}^{j} \mu_{i,k} < \frac{\alpha}{2} \sum_{j=1}^{n} \mu_{i,j} ; \qquad (17)$$

$$PI_{ci}^{U} = e_j \quad j : \sum_{k=1}^{j} \mu_{i,k} > (1-\frac{\alpha}{2}) \sum_{j=1}^{n} \mu_{i,j} , \qquad (18)$$

where: j is the index of the sorted data point that satisfies the corresponding inequalities; e_j denotes historical residuals of sorted data point j; and $\mu_{i,j}$ is membership grade of data point j to cluster i.

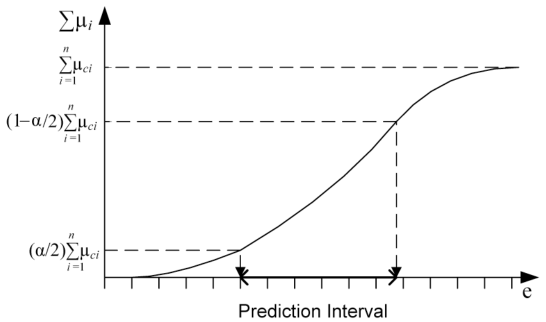

Fig. 7. PI calculation for one cluster

In the third step, the PI for each tcraining data point is calculated using the weighted mean of PI of each cluster:

$$PI_{j}^{L} = \sum_{i=1}^{c} \mu_{i,j} \times PI_{ci}^{L} ; \qquad (19)$$

$$PI_{j}^{U} = \sum_{i=1}^{c} \mu_{i,j} \times PI_{ci}^{U} , \qquad (20)$$

where: PI_{j}^{L} and PI_{j}^{U} are lower and upper prediction intervals for data point j.

Finally, prediction limits of EAC for each data point are computed as follows:

$$EAC_{j}^{L} = y_i + PI_{j}^{L} ; \qquad (21)$$

$$EAC_{j}^{U} = y_i + PI_{j}^{U} , \qquad (22)$$

where: EAC_{j}^{L} and EAC_{j}^{U} are lower and upper prediction limits of EAC for input data j.

3.5. LS-SVM for inference of EAC prediction limits

Once the prediction limits for each training data point are obtained, LS-SVM is utilized to establish two regression functions that model the relationship between the input data and its corresponding prediction limits. Tuning parameters of LS-SVM in this step are also selected via DE-based cross-validation. When the training process finishes, the model is then capable of estimating lower and upper prediction limits for new instances of input data.

3.6. Interval estimation of project cost at completion

In this step, the final model outputs (EAC_p , EAC_{j}^{L} , and EAC_{j}^{U}) are presented. The interval estimation of total cost is available for decision-making process. The planners or managers can anticipate the cost required to complete the project associated with uncertainty described in the form of prediction intervals.

4. Simulation result and comparisons

After the training process, the proposed model, EAC-LSPIM, is utilized to predict two testing projects (12 and 13). Projects 12 and 13 consist of 27 and 17 completion periods, respectively. To achieve interval forecast of EAC, the level of confidence is set as 95%, which is corresponding to α of 5%. In order to evaluate the accuracy of EAC point estimation, Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE) are employed. In addition, to assess the performance of EAC interval estimation, PICP and MPI are utilized.

Prediction results of EAC-LSPIM for two testing projects are illustrated in Tables 5 and 6, and Figures 8 and 9. In these tables and figures, EAC_A denotes the actual EAC. Meanwhile, EAC_L, EAC_P, and EAC_U represent lower prediction limit, point estimation and upper prediction limit of EAC, respectively. Deviation is the error between point estimate of EAC and the actual EAC.

In the experiment in which projects 12 and 13 serve as testing cases, the RMSE, MAPE, and MAE of point estimate are 0.044, 3,741, and 0.034, respectively. The PICP and the MPI derived from EAC-LSPIM are 97.73% and 19.22, respectively. Since the level of confidence is set as 95%, the derived PICP is desirable; and this demonstrates the reliability of the prediction results. Meanwhile, it can be observed that the width of PIs yielded by the proposed model is acceptable. On average, the range of predicted EAC is 19.22%, and this is relatively satisfactory in an operational environment of construction industry, which is often hazarded by uncertainty.

Table 5. Interval Estimation of EAC for project 12

No.	EAC_A	EAC_L	EAC_P	EAC_U	Deviation
1	91.69	74.45	84.79	93.77	6.90
2	91.69	69.06	79.41	88.39	12.28
3	91.69	79.11	89.33	98.31	2.36
4	91.69	78.35	88.74	97.85	2.95
5	91.69	76.39	86.75	95.73	4.94
6	91.69	73.08	83.34	92.39	8.35
7	91.69	73.86	84.01	93.11	7.68
8	91.69	73.02	82.82	91.71	8.87
9	91.69	78.20	87.94	96.78	3.75
10	91.69	82.36	92.59	101.57	0.90
11	91.69	80.73	91.00	99.98	0.69
12	91.69	80.89	91.26	100.29	0.43
13	91.69	80.14	90.55	99.57	1.14
14	91.69	84.09	94.48	103.46	2.79
15	91.69	80.85	91.25	100.22	0.44
16	91.69	83.75	94.12	103.09	2.43
17	91.69	80.40	90.74	99.72	0.95
18	91.69	80.69	91.03	100.01	0.66
19	91.69	75.33	85.69	94.68	6.00
20	91.69	77.65	88.02	97.00	3.67
21	91.69	78.10	88.48	97.47	3.21
22	91.69	80.08	90.47	99.46	1.22
23	91.69	77.24	87.62	96.61	4.07
24	91.69	82.37	92.73	101.71	1.04
25	91.69	82.57	92.48	101.29	0.79
26	91.69	82.12	91.70	100.44	0.01
27	91.69	83.60	93.24	102.06	1.55

Table 6. Interval Estimation of EAC for project 13

No.	EAC_A	EAC_L	EAC_P	EAC_U	Deviation
1	92.49	83.98	94.33	103.31	1.84
2	92.49	86.85	97.42	106.39	4.93
3	92.49	78.87	89.27	98.25	3.22
4	92.49	78.16	88.54	97.52	3.95
5	92.49	76.97	87.32	96.30	5.17
6	92.49	83.88	94.24	103.22	1.75
7	92.49	90.91	101.26	110.24	8.77
8	92.49	91.51	101.82	110.79	9.33
9	92.49	86.05	95.73	104.49	3.24
10	92.49	81.05	90.32	98.91	2.17
11	92.49	80.64	90.69	99.54	1.80
12	92.49	84.49	94.79	104.04	2.30
13	92.49	79.70	90.23	99.39	2.26
14	92.49	78.55	89.39	98.61	3.10
15	92.49	77.98	88.68	97.90	3.81
16	92.49	81.06	91.12	100.08	1.37
17	92.49	84.62	94.86	103.97	2.37

In order to validate the superiority of EAC-LSPIM, its performance is compared to other benchmarked approaches. It is noted that the newly developed model is composed of LS-SVM, MLIE, and DE-based cross-validation. In order to validate the superiority of the proposed prediction model, various machine learning techniques, namely M5-MT, ANN, and LS-SVM, has been integrated with MLIE and are applied to for interval prediction of EAC. For LS-SVM, the selection of tuning parameters is achieved via the grid search approach (Suykens *et al.* 2002; Shu *et al.* 2010). Utilizing this approach, various pairs of (γ and δ) are tried and the one with the best cross-validation accuracy is chosen. Accordingly, the values of γ and σ obtained from the grid search method, for point estimation of EAC, are 256 and 2.8, respectively. Meanwhile, the optimal values of γ and σ found by DE are 251.4 and 3.9, respectively.

The result comparison is shown in the Table 7. From Table 7, it is observable that the proposed model, EAC-LSPIM, has achieved the best result in point estimate of EAC having the smallest RMSE, MAPE, and MAE of testing data. Moreover, the model also yields the most desirable performance in interval estimation of project cost. Its prediction interval has the highest PICP value (97.73%) with relatively narrow MPI (19.22) compared to other outcomes.

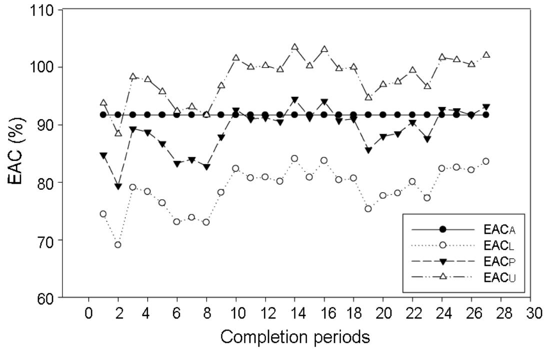

Fig. 8. EAC-LSMLPI prediction for project 12

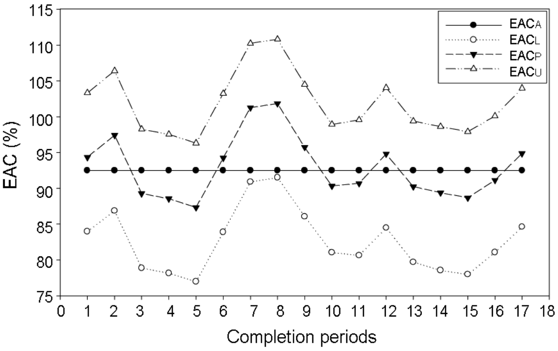

Fig. 9. EAC-LSMLPI prediction for project 13

To better demonstrate the performance of each benchmark model, the results of different combinations of 13 projects for training and testing have been added. In this experiment, 13 cases of experiment are carried out. In each case, a project serves as a testing set, the rest are training sets. As shown in Table 8, based on the average prediction results, the proposed model achieves the most desirable outcome. For point estimation, the RMSE, MAPE, and MAE of EAC-LSPIM are 3.812, 0.035, and 0.042, respectively. Meanwhile, for interval estimation, the PICP and MPI of the proposed model are 98.43% and 21.89. Observably, EAC-LSPIM is the most accurate model in point forecast of project cost. Moreover, it also achieves the highest PICP value corresponding to a relatively small value of MPI. These facts strongly proved the superiority of the new model over other benchmark approaches.

Table 7. Prediction result comparison for 2 testing projects

Prediction model		M5-MT	ANN	LS-SVM	EAC-LSPIM
Training	MAPE	3.16	3.11	1.81	2.36
	MAE	0.03	0.03	0.02	0.02
	RMSE	0.04	0.05	0.03	0.03
Testing	MAPE	7.63	3.83	3.90	3.74
	MAE	0.07	0.04	0.04	0.03
	RMSE	0.09	0.06	0.05	0.04
Interval Estimation	PICP	81.8	90.9	93.2	97.7
	MPI	27.9	23.0	20.8	19.2

Conclusion

This study proposes a new prediction model, namely EAC-LSPIM, to assist project manager in construction cost planning and monitoring. To address the uncertainty in construction cost forecasting, this study incorporates LS-SVM, MLIE, and DE to achieve interval forecasting of construction project cost.

In EAC-LSPIM, the utilization of LS-SVM is twofold. First, LS-SVM is used to infer the underlying function between input data and point estimation of ETC. Second, it is employed to model the mapping relationship between the input data and the prediction limits of EAC.

Moreover, by using MLIE, the new model derives the prediction interval by evaluating the uncertainty inherent in the data set, without any assumption or prior knowledge about model's error distribution.

In order to avoid over-fitting, our study employs DE search engine in the cross-validation process. The DE-based cross-validation successfully identifies the most appropriate set of tuning parameters and eliminates the need of expertise or trial-and-error process in parameter setting.

Consequently, the proposed model has the capacity to operate automatically without human intervention and domain knowledge. In addition, simulation and performance comparison have demonstrated the accuracy, the reliability, and the usability of EAC-LSPIM prediction. Therefore, the newly established model has a great potential to assist decision-makers in the field of construction management.

References

Abba, W. F. 1997. Earned value management: reconciling government and commercial practices, *Program Manager* 26(1): 58–63.

An, S.-H.; Park, U. Y.; Kang, K. I.; Cho, M. Y.; Cho, H. H. 2007. Application of support vector machines in assessing conceptual cost estimates, *Journal of Computing in Civil Engineering* 21(4): 259–264. http://dx.doi.org/10.1061/(ASCE)0887-3801(2007)21:4(259)

Bao, Y. K.; Liu, Z. T.; Guo, L.; Wang, W. 2005. Forecasting stock composite index by fuzzy Support Vector Machine regression, in *Proc. of the 4th International Conference on Machine Learning and Cybernetics,* 18–21 August, 2005, Guangzhou, China, Vol. 6: 3535–3540. http://dx.doi.org/10.1109/ICMLC.2005.1527554

Bezdek, J. C. 1981. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers Norwell, MA, USA. 256 p. http://dx.doi.org/10.1007/978-1-4757-0450-1

Table 8. Prediction result comparison for 13 projects

Model		Project													Average
		1	2	3	4	5	6	7	8	9	10	11	12	13	
M5-MT	MAPE	4.3	10.6	9.7	7.3	1.7	6.3	5.0	4.9	8.1	8.0	10.3	6.8	8.2	7.0
	MAE	0.0	0.1	0.1	0.1	0.0	0.1	0.0	0.0	0.1	0.1	0.1	0.1	0.1	0.1
	RMSE	0.1	0.1	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1	0.1	0.1	0.1	0.1
	PICP	93.1	79.2	90.0	88.0	100.0	95.0	86.4	96.3	94.4	87.1	80.7	85.2	82.4	89.1
	MPI	21.2	22.1	27.5	26.4	14.0	24.1	21.1	25.2	36.5	26.4	27.4	28.4	25.0	25.0
ANN	MAPE	7.2	7.7	6.4	4.5	4.1	6.5	3.2	7.6	5.6	3.1	8.8	4.7	5.7	5.8
	MAE	0.1	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.1	0.1
	RMSE	0.1	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1	0.1	0.1
	PICP	89.7	100.0	100.0	100.0	84.6	85.0	100.0	92.6	88.9	100.0	93.6	92.6	88.2	93.5
	MPI	28.5	31.6	24.1	23.1	25.6	28.6	30.4	22.7	25.4	27.1	30.7	23.4	21.5	26.4
LSSVM	MAPE	4.0	4.0	4.0	6.4	5.3	7.3	6.6	3.9	5.5	6.8	5.6	5.0	4.3	5.3
	MAE	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.1
	RMSE	0.1	0.0	0.0	0.1	0.1	0.1	0.1	0.0	0.1	0.1	0.1	0.1	0.1	0.1
	PICP	96.6	100.0	100.0	100.0	84.6	100.0	95.5	100.0	83.3	87.1	96.8	92.6	94.1	94.7
	MPI	21.6	22.0	26.2	34.0	19.7	36.4	32.1	35.6	22.8	24.6	21.4	22.4	22.1	26.2
EAC-LSPIM	MAPE	3.5	3.7	4.3	6.5	2.8	6.7	1.7	4.2	3.9	2.2	2.5	3.6	3.9	3.8
	MAE	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	RMSE	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	PICP	96.6	100.0	100.0	100.0	92.3	100.0	100.0	100.0	94.4	100.0	100.0	96.3	100.0	98.4
	MPI	17.8	24.3	16.6	25.1	17.7	29.1	20.2	28.1	20.8	21.2	25.4	19.0	19.3	21.9

Bhattacharya, B.; Solomatine, D. P. 2005. Neural networks and M5 model trees in modelling water level–discharge relationship, *Neurocomputing* 63: 381–396. http://dx.doi.org/10.1016/j.neucom.2004.04.016

Bishop, C. 2006. *Pattern recognition and machine learning*. Singapore: Springer Science+Business Media, LLC.

Bonakdar, L.; Etemad-Shahidi, A. 2011. Predicting wave run-up on rubble-mound structures using M5 model tree, *Ocean Engineering* 38(1): 111–118. http://dx.doi.org/10.1016/j.oceaneng.2010.09.015

Breiman, L.; Breiman, J.; Charles, J. S.; Olshen, R. A. 1984. *Classification and regression trees.* Chapman and Hall/CRC.

Chen, A.-L.; Wang, M.-L.; Liu, K. 2005. Prediction of the flow stress for 30 MnSi steel using evolutionary least squares support vector machine and mathematical models, in *Proc. of the IEEE International Conference on Industrial Technology ICIT,* 14–17 December, 2005, Hong Kong, China, 8968302.

Chen, T. L. 2008. *Estimate at completion for construction projects using evolutionary fuzzy neural inference model:* MS Thesis. Department of Construction Engineering, National Taiwan University of Science and Technology.

Cheng, M.-Y.; Peng, H.-S.; Wu, Y.-W.; Chen, T.-L. 2010. Estimate at completion for construction projects using evolutionary support vector machine inference model, *Automation in Construction* 19(5): 619–629. http://dx.doi.org/10.1016/j.autcon.2010.02.008

Cheng, M.-Y.; Roy, A. F. V. 2011. Evolutionary fuzzy decision model for cash flow prediction using time-dependent support vector machines, *International Journal of Project Management* 29(1): 56–65. http://dx.doi.org/10.1016/j.ijproman.2010.01.004

Christensen, D. S. 1993. Determining an accuracy estimate at completion, *National Contract Management Journal* 25: 17–25.

Christensen, D. S.; Antolini, R. C.; McKinney, J. W. 1995. A review of estimate at completion research, *Journal of Cost Analysis and Management,* 41–62.

De Brabanter, K.; De Brabanter, J.; Suykens, J. A. K.; De Moor, B. 2011. Approximate confidence and prediction intervals for least squares support vector regression, *IEEE Transactions on Neural Networks* 22(1): 110–120. http://dx.doi.org/10.1109/TNN.2010.2087769

De Brabanter, K.; Karsmakers, P.; Ojeda, F.; Alzate, C.; De Brabanter, J.; Pelckmans, K.; De Moor, B.; Vandewalle, J.; Suykens, J. A. K. 2010. *LS-SVMlab Toolbox User's Guide version 1.8*. Internal Report 10-146, ESAT-SISTA, K.U. Leuven (Leuven, Belgium).

Gestel, T. V.; Suykens, J. A. K.; Baesens, B.; Viaene, S.; Vanthienen, J.; Dedene, G.; De Moor, B.; Vandewalle, J. 2004. Benchmarking least squares support vector machine classifiers, *Machine Learning* 54(1): 5–32. http://dx.doi.org/10.1023/B:MACH.0000008082.80494.e0

Guo, Z.; Bai, G. 2009. Application of least squares support vector machine for regression to reliability analysis, *Chinese Journal of Aeronautics* 22(2): 160–166. http://dx.doi.org/10.1016/s1000-9361(08)60082-5

Hegazy, T.; Ayed, A. 1998. Neural network model for parametric cost estimation of highway projects, *Journal of Construction Engineering and Management* 124(3): 210–218. http://dx.doi.org/10.1061/(asce)0733-9364(1998)124: 3(210)

Heskes, T. 1997. Practical confidence and prediction interval, *Advances in Neural Information Processing Systems 9.* Cambridge: MIT Press. 176–182. http://dx.doi.org/10.1.1.56.3753

Huang, Z.; Chen, H.; Hsu, C. J.; Chen, W. H.; Wu, S. 2004. Credit rating analysis with support vector machines and neural networks: a market comparative study, *Decision Support System* 37(4): 543–558. http://dx.doi.org/10.1016/S0167-9236(03)00086-1

Iranmanesh, H.; Mojir, N.; Kimiagari, S. 2007. A new formula to "Estimate at Completion" of a project's time to improve "Earned Value Management System", in *Proc. of the IEEE International Conference on Industrial Engineering and Engineering Management*, 2–4 December, 2007, Singapore, 1014–1017.

Jekabsons, G. 2010. *M5 regression tree and model tree toolbox for Matlab.* Technical Report, Institute of Applied Computer Systems, Riga Technical University.

Kaluzny, B. L.; Barbici, S.; Berg, G.; Chiomento, R.; Derpanis, D.; Jonsson, U.; Shaw, R. H. A. D.; Smit, M. C.; Ramaroson, F. 2011. An application of data mining algorithms for shipbuilding cost estimation, *Journal of Cost Analysis and Parametrics* 4(1): 2–30. http://dx.doi.org/10.1080/1941658x.2011.585336

Khosravi, A.; Nahavandi, S.; Creighton, D. 2010. Construction of optimal prediction intervals for load forecasting problems, *IEEE Transactions on Power Systems* 25(3): 1496–1503. http://dx.doi.org/10.1109/TPWRS.2010.2042309

Khosravi, A.; Nahavandi, S.; Creighton, D.; Atiya, A. F. 2011. Lower upper bound estimation method for construction of neural network-based prediction intervals, *IEEE Transactions on Neural Network* 22(3): 337–346. http://dx.doi.org/10.1109/TNN.2010.2096824

Kim, G.-H.; An, S.-H.; Kang, K.-I. 2004. Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning, *Building and Environment* 39(10): 1235–1242. http://dx.doi.org/10.1016/j.buildenv.2004.02.013

Kiranyaz, S.; Ince, T.; Yildirim, A.; Gabbouj, M. 2009. Evolutionary artificial neural networks by multi-dimensional particle swarm optimization, *Neural Networks* 22(10): 1448–1462. http://dx.doi.org/10.1016/j.neunet.2009.05.013

Kong, F.; Wu, X.-J.; Cai, L.-Y. 2008. A novel approach based on support vector machine to forecasting the construction project cost, in *Proc. of the International Symposium on Computational Intelligence and Design*, 17–18 October, 2008, Wuhan, China, 1045–1076.

Lam, J. P.; Veall, M. R. 2002. Bootstrap prediction intervals for single period regression forecasts, *International Journal of Forecasting* 18(1): 125–130. http://dx.doi.org/10.1016/s0169-2070(01)00112-1

Lam, K. C.; Palaneeswaran, E.; Yu, C.-Y. 2009. A support vector machine model for contractor prequalification, *Automation in Construction* 18(3): 321–329. http://dx.doi.org/j.autcon.2008.09.007

Yu, L.; Chen, H.; Wang, S.; Lai, K. K. 2009. Evolving least squares support vector machines for stock market trend mining, *IEEE Transactions on Evolutionary Computation* 13(1): 87–102. http://dx.doi.org/10.1109/TEVC.2008.928176

Liu, L.; Zhu, K. 2007. Improving cost estimates of construction projects using phased cost factors, *Journal of Construction Engineering and Management* 133(1): 91–95. http://dx.doi.org/10.1061/(asce)0733-9364(2007)133: 1(91)

Lu, C.-J.; Lee, T.-S.; Chiu, C.-C. 2009. Financial time series forecasting using independent component analysis and support vector regression, *Decision Support Systems* 47(2): 115–125. http://dx.doi.org/10.1016/j.dss.2009.02.001

Hongwei, M. 2009. An improved support vector machine based on rough set for construction cost prediction, in *Proc. of the International Forum on Computer Science-Technology and Applications,* 25–27 December, 2009, Chongqing, China, Vol. 2: 3–6. http://dx.doi.org/10.1109/IFCSTA.2009.123

Mencar, C.; Castellano, G.; Fanelli, A. M. 2005. Deriving prediction intervals for neuro-fuzzy networks, *Mathematical and Computer Modelling* 42(7–8): 719–726. http://dx.doi.org/10.1016/j.mcm.2005.09.001

Nassar, K. M.; Nassar, W. M.; Hegab, M. Y. 2005. Evaluating cost overruns of asphalt paving project using statistical process control methods, *Journal of Construction Engineering and Management* 131(11): 1173–1178. http://dx.doi.org/10.1061/(ASCE)0733-9364(2005) 131:11(1173)

Nasseri, M.; Asghari, K.; Abedini, M. J. 2008. Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network, *Expert Systems with Applications* 35(3): 1415–1421. http://dx.doi.org/10.1016/j.eswa.2007.08.033

Olive, D. J. 2007. Prediction intervals for regression models, *Computational Statistics & Data Analysis* 51(6): 3115–3122. http://dx.doi.org/10.1016/j.csda.2006.02.006

Oliveira, J. V. D.; Pedrycz, W. 2007. *Advances in fuzzy clustering and its applications*. John Wiley & Sons Ltd. 434 p. http://dx.doi.org/10.1002/9780470061190

Price, K. V.; Storn, R. M.; Lampinen, J. A. 2005. *Differential evolution a practical approach to global optimization.* Berlin, Heidelberg: Springer-Verlag.

Razi, M. A.; Athappilly, K. 2005. A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree (CART) models, *Expert Systems with Applications* 29(1): 65–74. http://dx.doi.org/10.1016/j.eswa.2005.01.006

Samarasinghe 2006. *Neural networks for applied sciences and engineering.* USA: Taylor & Francis Group, LLC.

Samui, P.; Kothari, D. P. 2011. Utilization of a least square support vector machine (LSSVM) for slope stability analysis, *Scientia Iranica* 18(1): 53–58. http://dx.doi.org/10.1016/j.scient.2011.03.007

Shrestha, D. L.; Solomatine, D. P. 2006. Machine learning approaches for estimation of prediction interval for the model output, *Neural Networks* 19(2): 225–235. http://dx.doi.org/10.1016/j.neunet.2006.01.012

Shu, C. W.; Chang, C. C.; Lin, C. J. 2010. *A practical guide to support vector classification.* Technical Report. Department of Computer Science, National Taiwan University.

Sonmez, R. 2011. Range estimation of construction costs using neural networks with bootstrap prediction intervals, *Expert Systems with Applications* 38(8): 9913–9917. http://dx.doi.org/10.1016/j.eswa.2011.02.042

Stine, R. A. 1985. Bootstrap prediction intervals for regression, *Journal of the American Statistical Association* 80(392): 1026–1031. http://dx.doi.org/10.2307/2288570

Storn, R.; Price, K. 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* 11(4): 341–359. http://dx.doi.org/10.1023/A:1008202821328

Suykens, J.; Gestel, J. V.; Brabanter, J. D.; Moor, B. D.; Vandewalle, J. 2002. *Least square support vector machines*. Singapore: World Scientific Publishing Co. Pte. Ltd.

Timofeev, R. 2004. *Classification and Regression Trees (CART): theory and applications*: Master's Thesis. Center of Applied Statistics and Economics, Humboldt University, Berlin.

Trost, S. M.; Oberlender, G. D. 2003. Predicting accuracy of early cost estimates using factor analysis and multivariate regression, *Journal of Construction Engineering and Management* 129(2): 198–204. http://dx.doi.org/10.1061/(asce)0733-9364(2003)129: 2(198)

Wang, H.; Hu, D. 2005. Comparison of SVM and LS-SVM for regression, in *Proc. of the International Conference on Neural Networks and Brain (ICNNB),* 13–15 October, 2005, Beijing, China, 279–283. http://dx.doi.org/10.1109/ICNNB.2005.1614615

Witten, I. H.; Frank, E. 2000. *Practical machine learning tools and techniques with java implementations*. USA: Morgan Kaufmann.

Wong, B. K.; Bodnovich, T. A.; Selvi, Y. 1997. Neural network applications in business: a review and analysis of the literature (1988–1995), *Decision Support System* 19(4): 301–320. http://dx.doi.org/10.1016/s0167-9236(96)00070-x

Wonnacott, T. H.; Wonnacott, R. J. 1996. *Introductory statistics*. New York: Wiley.

Xie, X. L.; Beni, G. 1991. A validity measure for fuzzy clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(8): 841–847. http://dx.doi.org/10.1109/34.85677

Zhang, J. R.; Zhang, J.; Lok, T. M.; Lyu, M. R. 2007. A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied Mathematics and Computation* 185(2): 1026–1037. http://dx.doi.org/10.1016/j.amc.2006.07.025

Zhu, W.-J.; Feng, W.-F.; Zhou, Y.-G. 2010. The application of genetic fuzzy neural network in project cost estimate, in *Proc. of the International Conference on E-Product E-Service and E-Entertainment (ICEEE),* 1–4 November, 2010, Henan, China, 1–4. http://dx.doi.org/10.1109/ICEEE.2010.5660115

Min-Yuan CHENG is currently a Professor at the Department of Construction Engineering, National Taiwan University of Science and Technology. He holds lectures in Construction Automation and Construction Process Re-engineering. He has published many papers in various international journals such as *Automation in Construction*, *Journal of Construction Engineering and Management*, and *Expert Systems with Applications*. His research interests include management information system, applications of artificial intelligence, and construction management process reengineering.

Nhat-Duc HOANG is currently a lecturer at Department of Technology and Construction Management, Faculty of Building and Industrial Construction, National University of Civil Engineering, Hanoi, Vietnam. He got the MSc and PhD degrees at National Taiwan University of Science and Technology, Taipei, Taiwan. His research focuses on applications of Artificial Intelligence in Construction engineering and management. His research articles have been published in *Journal of Computing in Civil Engineering* (ASCE), *Engineering Applications of Artificial Intelligence*, and *Automation in Construction.*