www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

USING PARTICLE SWARM OPTIMIZATION TO PREDICT COST CONTINGENCY ON TRANSPORTATION CONSTRUCTION PROJECTS

SUBMITTED: May 2016
REVISED: November 2016
PUBLISHED: December 2016 at http://www.itcon.org/2016/30
EDITOR: Amor R.

Sang C. Lhee, General Manager,
Technology and Research Institute, Dongwoo Engineering and Construction, Seoul, S Korea;
lheesch@naver.com

Raja R.A. Issa, Holland Professor,
Rinker School of Construction Management, University of Florida, Gainesville, FL;
raymond-issa@ufl.edu

Ian Flood, Holland Professor,
Rinker School of Construction Management, University of Florida, Gainesville, FL;
flood@ufl.edu

SUMMARY: *Cost Contingency is a financially important amount added to the base estimate for covering unforeseen uncertainties and risks in construction projects, including construction difficulties, design changes during construction, and inaccuracies in the estimating process. An accurate cost contingency is critical to construction project participants having a significant impact on project financial successes and other organizational activities. This study proposes a new approach to predicting the owner's cost contingency on transportation construction projects using particle swarm optimization (PSO), a population-based stochastic optimization technique inspired by the social behavior of flocking birds or schooling fish. Through a comparison of performance with an artificial neural network (ANN) based approach using historical data from Florida Department of Transportation (FDOT) construction projects, the findings indicated PSO more accurately predicts the owner's cost contingency.*

KEYWORDS: *Cost contingency, Risk and Uncertainty; Owners perspective; Transportation construction project; Particle swarm optimization; Artificial neural networks.*

REFERENCE: *Sang C. Lhee, Raja R.A. Issa, Ian Flood (2016). Using particle swarm optimization to predict cost contingency on transportation construction projects. Journal of Information Technology in Construction (ITcon), Vol. 21, pg. 504-516, http://www.itcon.org/2016/30*

1. INTRODUCTION

Cost contingency is defined as an amount of money added to the base estimated amount in order to account for substantial uncertainties in quantities, unit costs, and the possibility of unforeseen risk events related to quantities, work elements, and other project requirements (Molenaar 2005). Contingency helps construction project participants carry out financially successful projects, providing the flexibility required to cope with uncertainties and risks that occur during all construction phases that threaten to achieve organizational objectives. The accurate prediction of contingency and its adequacy is significant to the financial success of projects and has tremendous impact on project outcomes for all participants (Dey et al. 1994; Baccarini 2004). For example, a high contingency may result in ineffective cost management, uneconomic completion of projects, and a lack of available funds for other organizational activities, while a low contingency may cause problems such as inadequate funding, an unrealistic financial environment, and unsatisfactory performance outcomes for projects. It is important therefore that an accurate contingency is allocated to each estimate to facilitate optimal project performance and financial resource usage.

Currently, construction participants tend to allocate contingency intuitively rather than systematically (Gunhan and Arditi 2007). For a long time the common practice for allocating contingency in the construction industry has been to add a single predetermined percentage from the overall project contract amount through conjecture, intuition, and past experience. Modern estimating textbooks represent contingency as a fixed percentage of the estimated contract amount and the suggested contingency is generally around 5-10% of the contract amount (Smith and Bohn 1999). Although this traditional predetermined percentage method is simple and easy to use, it has several weaknesses (Thompson and Perry 1992; Karlsen and Lereim 2005): (1) it is overly simplistic and heavily dependent on the faith of estimators in their own experience; (2) the percentage reported is arbitrarily estimated and not appropriate for a specific project; (3) there may be a tendency to double count risks; and (4) it is still a single figure implying a degree of certainty that is simply not justified.

2. BACKGROUND

Several alternative contingency prediction models have been proposed in the literature. Mak and Picken (2000) proposed a method they called "Estimating using Risk Analysis (ERA)" to substantiate contingency by identifying uncertainties and estimating their financial implications. Oberlender and Trost (2001) developed the Estimate Score Program (ESP) software to predict contingency and to assess the accuracy of cost estimates using historical cost data. Touran (2003) proposed a probabilistic contingency prediction model that considers the expected number of change orders and their impacts on project costs (i.e. the average cost of change orders). Moselhi et al. (1993) developed a decision-support system that helps contractors prepare competitive bids for construction projects by estimating an optimal contingency in future bid situations using artificial neural network (ANN) methodology. Thal et al. (2010) suggested a multiple linear regression model to predict contingency amounts for military construction projects. Lhee et al. (2012) developed an easy-to-use ANN (artificial neural network) based contingency prediction tool for the Florida Department of Transportation (FDOT) construction projects by identifying potential input factors that affect contingency and finding an appropriate form of contingency Lhee et al. (2012) demonstrated the potential of intelligent computing approaches, specifically artificial neural networks, for predicting financial contingency for asphalt resurfacing projects. Baccarini and Love (2014) analyzed statistical characteristics of cost contingency in water infrastructure projects and then determined the best-fit probability distribution using the empirical distributions of cost contingency in order to improve the accuracy of a contingency estimate. Love et al. (2015) also suggested a contingency estimating method for road construction projects which were procured using lump-sum contracts, using the log-logistic probability density function to best model the behavior of cost overruns.

Particle swarm optimization (PSO) (Kennedy and Eberhart, 1995) is a population-based stochastic optimization technique for finding the best solution during the search process, inspired by the social behavior of flocking birds or schooling fish. Recently, PSO has been used as an optimization tool on a wide range of applications within the construction industry. Zhang et al. (2006) implemented PSO for a permutation-based scheme on resource-constrained project scheduling problems with the objective of minimizing project duration. Yang (2007) used PSO to find the complete time-cost profile over a set of feasible project durations and to facilitate bi-criterion time-cost trade-off analysis. Zhang and Wang (2008) proposed a PSO-based methodology to solve the construction site unequal-area facility layout problem. Dimou and Koumousis (2009) used PSO for the reliability-based optimal design of statistically determinate truss structures. Reddy and Adarsh (2010) proposed a swarm intelligence based methodology for optimal and reliable design of composite irrigation channels. Gopalakrishnan (2010) suggested a new hybrid approach for the back-calculation of flexible pavement layer moduli by integrating both ANN and PSO techniques. Ashuri and Tavakolan (2012) solved complex time-cost-resource optimization problems in construction project planning using a hybrid PSO approach. Yazdi et al. (2012) applied PSO for calibration of soil parameters used within a linear elastic-hardening plastic constitutive model. Chen et al. (2015) applied PSO in the form-finding analysis of a suspension bridge installation to overcome the conventional Newton-Raphson iterative method. However, the researchers found that PSO has seldom been applied to the solution of prediction problems. This study proposes and evaluates the application of PSO to the problem of predicting the owner's cost contingency on transportation construction projects. Using historical FDOT project data, the paper shows the viability of the PSO-based approach by comparing its performance with that of a proven ANN-based approach.

2.1 Particle Swarm Optimization

PSO simulates social behaviors such as bird flocking or fish schooling to determine a promising position for certain objectives in a multidimensional space (Kennedy and Eberhart 1995; Eberhart and Shi 2001). Similar to other evolutionary computation techniques such as Genetic Algorithm (GA), PSO conducts its search procedure using a population ("swarm") of individuals ("particles"). Each particle represents a potential solution to the problem like the phenotype in a GA and transitions through the problem space towards the current optimum particles. However, unlike GA, PSO does not need evolutionary operators such as crossover and mutation. Initialized with a group of random particles, PSO searches for an optimum solution or the best solution by updating successive generations. In every generation, each particle is updated by the following three best values: (i) *pbest* (personal best) is the best solution a particle has achieved so far; (ii) *lbest* (local best) is the best solution obtained so far by any particle among the neighbors of the particle; and (iii) *gbest* (global best) is the best solution from all members of the current population.

After finding the above three best values, the status of each particle within the search space is determined by updating its position and velocity according to the following two equations (Kennedy and Eberhart 1995). Equation (1) determines a particle's new position based on its personal best and global best positions. Equation (2) determines a particle's new velocity based on its previous velocity and the distance from its current positions to its personal best (*pbest*) and the global best (*gbest*) positions. In other words, at generation n, the position of the p-th particle ($X_p(n)$) in the search space is updated by the velocity ($V_p(n)$) from the position of the particle at the last generation, $X_p(n-1)$. The velocity of the p-th particle ($V_p(n)$) in the search space is determined by the position and velocity of the particle at the last generation $n-1$, $X_p(n-1)$ and $V_p(n-1)$, namely *pbest* and *gbest*.

$$X_p(n) = V_p(n) + X_p(n-1) \quad (1)$$

$$V_p(n) = w(t)V_p(n-1) + c_1 r_1 (PX_p - X_p(n-1)) + c_2 r_2 (GX - X_p(n-1)) \quad (2)$$

where $X_p(n)=\{x_{p1}(n), x_{p2}(n), \cdots, x_{pN}(n)\}$ denotes the N-dimensional position for the *p-th* particle in the *n-th* generation; $V_p(n)=\{v_{p1}(n), v_{p2}(n), \cdots, v_{pN}(n)\}$ denotes the N-dimensional velocity for the *pth* particle in the *n*th generation; $p=1, 2, \cdots, P$; P=population size; $n=1, 2, \cdots, N$; N=generation limit; PX_p denotes the personal best for the *p-th* particle (*pbest*); GX denotes the global best (*gbest*); c_1 and c_2 are positive constants (learning factors); r_1 and r_2 are uniformly distributed random numbers between 0 and 1; $w(t)$ is the inertia weight used to control the impact of the previous velocities on the current one.

Figure 1 shows the mechanism used by PSO for finding the new position of a particle during its search process. A swarm of particles, which is numbered 1 through 6, are located in a 3D search space. Particle 1 on the current best position ($X_p(n)$), flies toward its new best position ($X_p(n+1)$) based on its previous best position (*pbest*) and the best position among the swarm (particle 6, *gbest*).

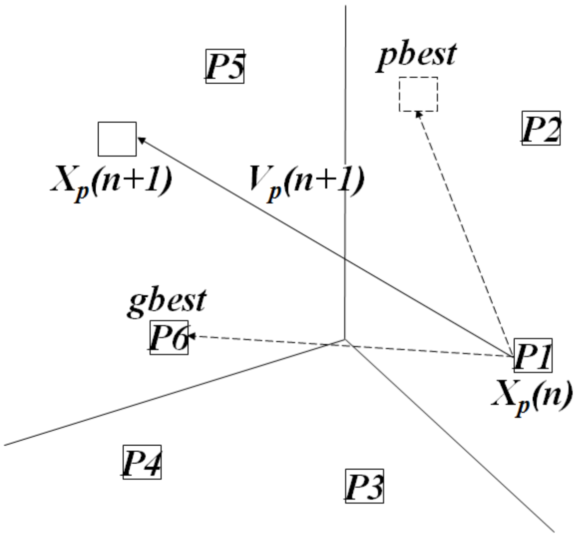

Figure 1. Search mechanism of the PSO-based method

3. METHODOLOGY

Figure 2 shows the basic PSO procedure for finding the best optimal position of a swarm. PSO shares many common features with GAs, such as randomly generating an initial population, searching for optimal solutions by updating from generation to generation, and using the fitness value to evaluate the population. GA reproduces chromosomes of the next generation from unclassified survivals using genetic operators like crossover and mutation.

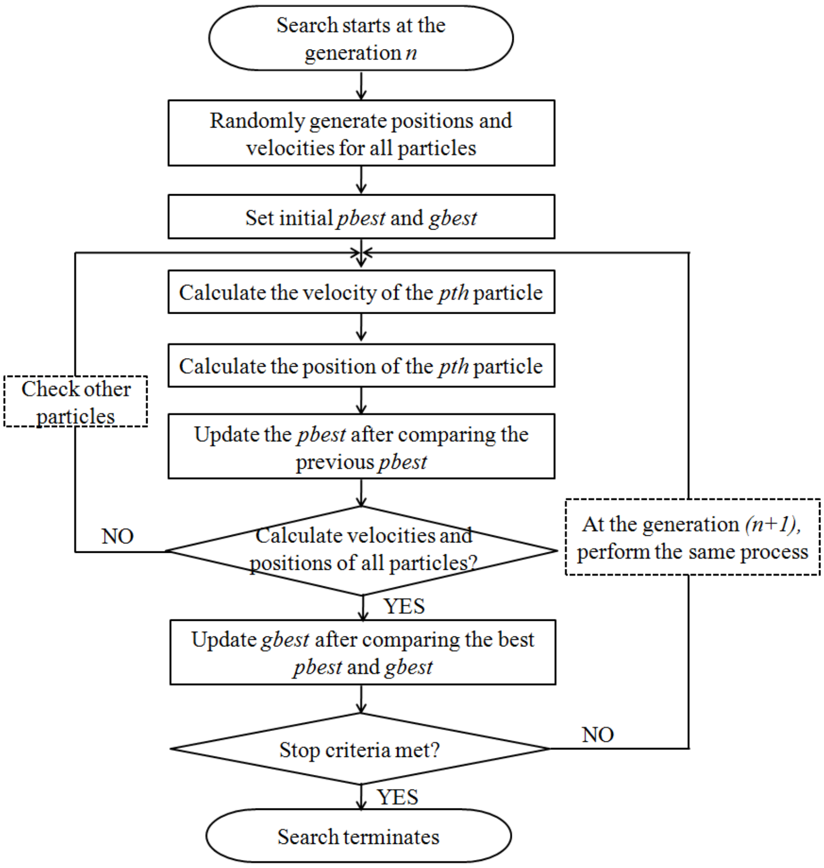

Figure 2. Procedure of the PSO-based method

However, in PSO, particles are updated with the internal velocity, the discoveries of the particles themselves, and the previous experiences of other neighbors (particles). PSO has some advantages over GAs namely its easy implementation, its faster search process, and its greater capability of escaping local optima (Eberhart and Shi 1998; Zhang et al. 2006). However, it must be noted that one cannot be proven to always outperform the other in terms of efficiency and effectiveness because GAs and PSO both rely on a stochastic search and the capabilities or performances of both methods is problem dependent (Wolpert and Macready 1997; Yang 2007).

3.1 Development of PSO-based Model for Predicting Contingency

Like procedures for developing empirical models such as artificial neural network (ANN) and regression analysis, the first step in the development of the PSO-based prediction model should be to establish the application objectives of the model (i.e. to determine the output variables, the focus of the model, and the required level of accuracy) and the input variables likely to be important (Flood and Issa 2010). Figure 3 represents the basic structure of the prediction model which consists of four input variables and one output variable. The contingency differential amount was adopted as the output variable, and is calculated as the difference between the original contract amount and final contract amount as proposed by Mak and Picken (2000). Four input variables were included (number of bidders, project year, project duration, and project amount) all of which have been determined to significantly influence the owner's contingency on transportation construction projects (see, for example, FHWA (2007) and Popescue et al. (2003)). These were also the four inputs used to predict contingency on transportation construction projects in the ANN-based study reported by Lhee et al. (2012) which is to be used as a benchmark for this study. In a low-bid wins award environment, the participation of many bidders will tend to lower the successful bid price through increased competition, but this situation also creates more risks and increases the chances of overrunning the project estimates. The second input variable, project year, is defined as the start year of the contract and is related to financial conditions and the number of other projects owners have during the construction phase. The third input variable, project duration, is important since projects with longer durations are more likely to include complex tasks and they have a higher risk of cost impacting factors such as unforeseen weather delays (e.g. in winter or rainy seasons). Finally, projects with larger contract amounts, the fourth input variable, tend to be more complex and more difficult in scope and thus are more likely to be exposed to significant risks against their original contract amounts.

As an implementation tool for developing the PSO-based prediction model, the NeuroShell Predictor software developed by the Ward System Group Inc. was used for this study. The software provides state-of-the-art prediction algorithms not only for artificial neural networks (ANNs), but also for particle swarm optimization (PSO) as training methods. The PSO training algorithm inside the software is developed from the basic swarm optimization theory based on Equations (1) and (2) as proposed by Eberhart and Kennedy (1995). The procedure would need to be rerun for different types of projects.

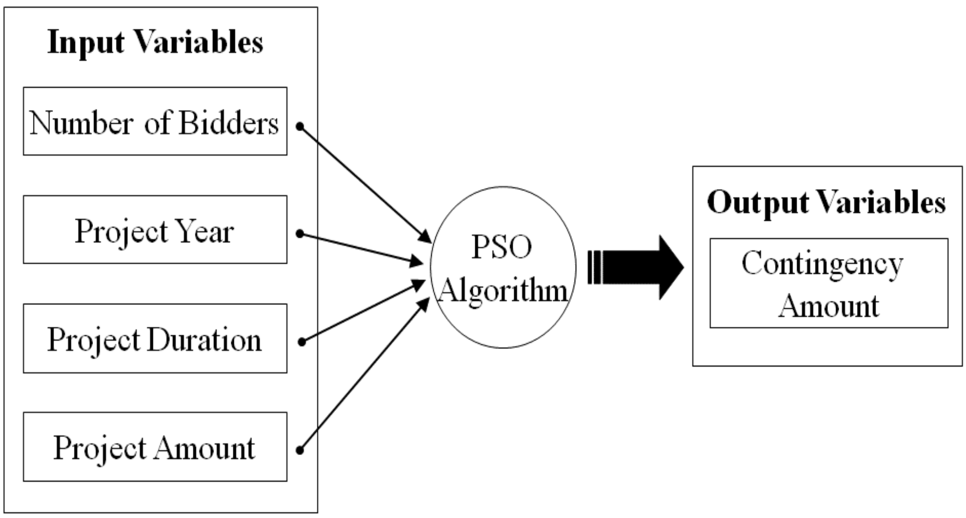

Figure 3. Basic structure of the PSO-based prediction model

3.2 Data Collection

The development and evaluation of the PSO-based prediction model was based on data collected from 492 FDOT projects completed from 2004 to 2006. The contingency-related data and corresponding information about the four input variables were retrieved from the FDOT quarterly time and cost reports. The datasets were randomly divided into three groups in the ratio of 60% to 20% to 20% for training, testing, and validation of the models respectively. The training data set, which contained data on 315 projects, was used as the target observations for development of the prediction model. The testing data set contained data on 86 projects and was used to evaluate and compare the performances of the developing models during the development stage. The last group, the validation data set, contained data on 91 projects and was used to make a concluding assessment and validation of the performance of the final model. The testing and validation data sets were prescreened to ensure that all output values were within the scope of those appearing within the training data set, a constraint imposed by the PSO.

FDOT projects use many alternative contracting methods such as the design-build delivery method, A+B (cost+time) bidding, the average bid method (BAM), and the incentive/disincentive contracting method (FDOT 2010) for improving the construction project and speeding up the project delivery without compromising safety or quality. According to the Major Project Program Cost Estimating Guidance (FHWA 2007), design-build contracts have shown little increase in construction costs from start to completion of projects under a negotiated contract amount and thus may require a smaller contingency amount. The magnitude of contingency may also be different depending on the bid award method type (Popescue et al. 2003). Categorical variables that do not have a natural progression in their values, such as delivery method and award type, are best represented by different models for each category (Flood and Issa, 2010). This study, therefore, focused on projects of the design-bid-build delivery type with a lowest bid wins award type.

4. RESULTS AND COMPARISONS

4.1 Training and Testing

The NeuroShell Predictor software provides the following net statistics for measuring the performance of the proposed model. In this study, the values of these net statistics were used to compare the performance of the PSO-based model to the ANN model in terms of the accuracy with which they can predict the owner's contingency amount.

- R-squared: the indicator for comparing the accuracy of the prediction model to that of a trivial benchmark model wherein the prediction is just the average of all example output values (the closer the R-squared value is to 1, the better the model is able to make predictions).
- Average error: the mean over all patterns of absolute differences between the actual output values and the predicted output values (i.e. the absolute value of the actual output values minus the predicted output values divided by the number of patterns).
- Correlation: the measure of how the actual output values and predicted output values correlate to each other in terms of direction (Pearson's correlation coefficient).

On implementing the PSO-based prediction model, the optimization goal was set to minimize the average error across all training patterns. Training was set to terminate when either of the following two conditions were met: (1) the total number of completed generations reached 250; or (2) the maximum number of generations since the last improvement reached 20. These cut-off criteria were found to correspond with minimal further improvement in most optimization runs as can be seen in Figure 4.

Figure 4 shows the progress in training for the PSO-based prediction model. The training algorithm stopped at the 243rd generation after meeting the second termination criteria. In order to show the viability of the PSO approach in this prediction problem, the performance of the model was compared with that of a previously developed ANN-based prediction model reported by Lhee et al. (2012). The ANN was structured with the same four input variables and output variable used for the PSO-based prediction model (see Figure 3). The ANN comprised a three-layer (single hidden layer) feedforward backpropagation (BP) network which is one of the most commonly used systems in engineering applications, and determines the connection weights using an error gradient descent technique.

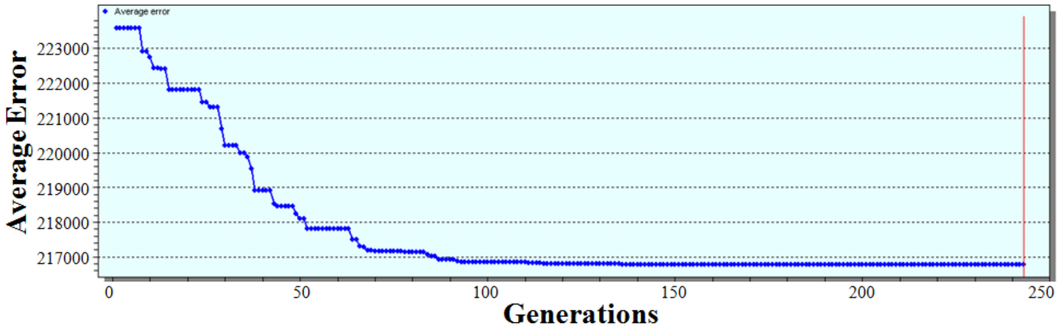

Figure 4. Finding the optimal generation during the training process for the PSO-based prediction model

Table 1 compares the performance of the two models (PSO and ANN) for predicting the contingency amount. Based on the values of the three net statistics, the PSO-based prediction model demonstrated significantly better performance than the ANN-based prediction model, both for the training and testing data sets. The performances measured by the R-squared values were improved by 31% on the training data set and by 54% on the testing data set. The values of average error were reduced by 448% on the training data set and by 165% on the testing data set. The performances from the values of correlation were also improved by 17% on the training data set and by 32% on the testing data set.

Table 1. Performance of Two Prediction Models on the Training and Testing Data Set

Data set	Net statistics	PSO-based model	ANN-based model
Training data set	R-squared	0.990	0.684
	Average error	$50,214.8	$275,290.0
	Correlation	0.995	0.827
Testing data set	R-squared	0.947	0.434
	Average error	$84,203	$223,518.8
	Correlation	0.975	0.660

Figure 5 is a scatter plot of the actual versus predicted contingency amounts for the two prediction models on the training data set. If the networks had learned all training patterns perfectly, then all points would fall on the 45° line shown. Likewise, Figure 6 shows the actual contingency amounts versus the predicted contingency amounts for the two models on the testing data set. More importantly, if the prediction models have been perfectly developed as a model which is able to generalize to patterns of the problems not used in training data set, then all points in Figure 5 would fall on the diagonal line indicated. As the higher correlation values on two data sets (respectively 0.990 and 0.947) indicated, all patterns for the PSO-based model were plotted around the 45° line. Based on the plots in Figures 4 and 5, it is apparent that the PSO-based prediction model provides consistently better performance than the ANN across the range of possible output values, and that there are no distinct outlier points.

In order to see whether the two prediction models under-predicted or over-predicted the contingency amount on the testing data set, the mean differences between the actual predicted output values were checked. The PSO-based prediction model predicted higher contingencies than the actual values for the testing data set having a mean difference of $29,664. Similarly, the ANN-based prediction model on the testing data set also predicted higher contingencies than the actual values having a mean difference of $16,799.

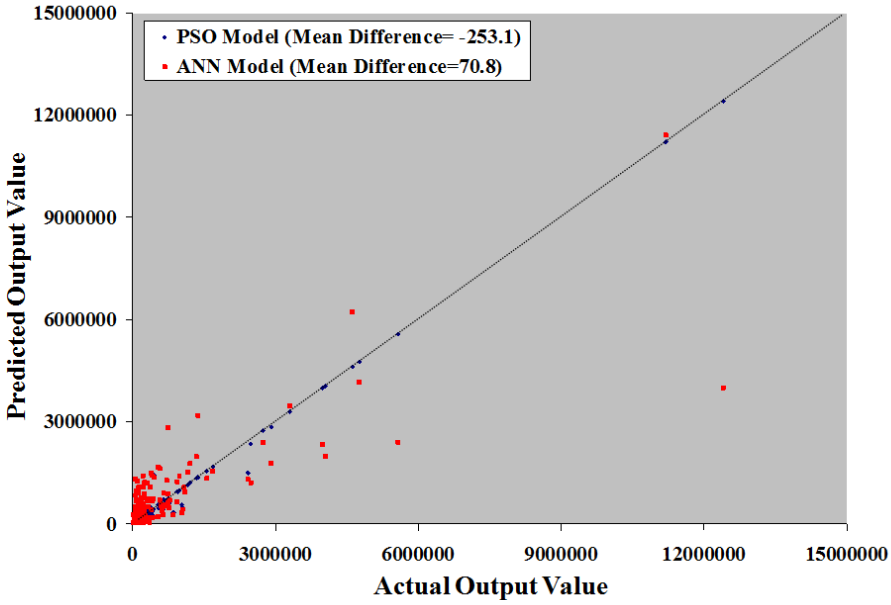

Figure 5. Correlation between actual and predicted output values for the training data set

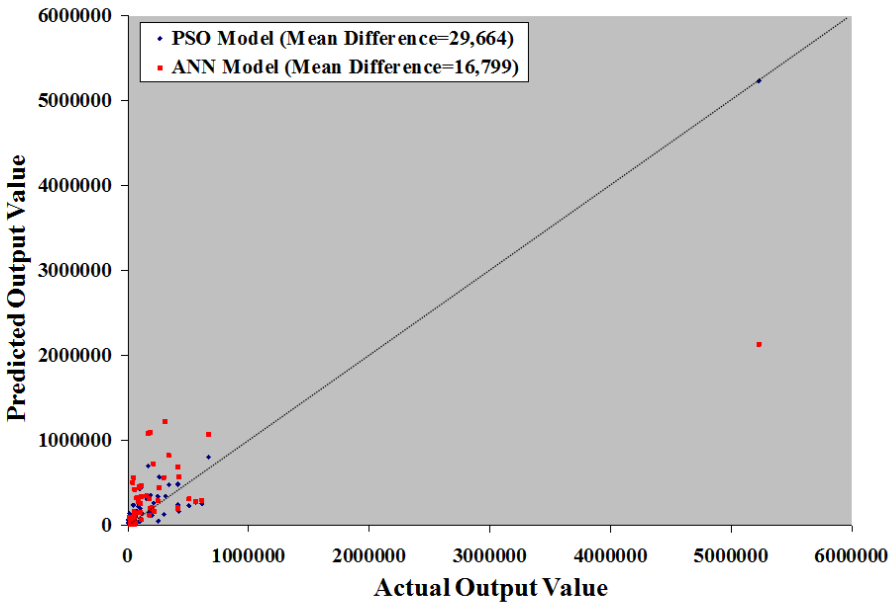

Figure 6. Correlation between actual and predicted output values for the testing data set

4.2 Final Validation

In order to provide a final assessment of the performances of the two prediction models and show the validity of the PSO approach in predicting the owner's contingency amount, it is necessary to reevaluate the selected best performing models using the validation data set. The purpose of this step is to remove any bias that the final models may have towards the data set (the testing data) that was used to select them. Table 2 shows the performance of the two prediction models on the validation data set. The PSO-based prediction model showed an improvement of 100% over the ANN-based model for the R-squared value, and of 45% in the correlation value. In addition, the PSO-based model was 61% lower than the ANN-based model for the average error. Therefore, it proved that the PSO approach is much better than the ANN approach on predicting the owner's contingency amount, providing higher performances of net statistics on the testing and validation data sets and showing its validity.

Table 2. Performance of Two Prediction Models on the Validation Data Set

Net statistics	Model type	
	PSO-based model	ANN-based model
R-squared	0.509	0.002
Average error	$137,647.6	$221,482.7
Correlation	0.748	0.412

Figure 7 shows the scatter plot for the predicted versus actual output value on the validation data set for the two prediction models. Visually, it appears that the data points for the PSO-based model are more closely clustered around the 45° line than those for the ANN-based model, and there were the fewer outliers.

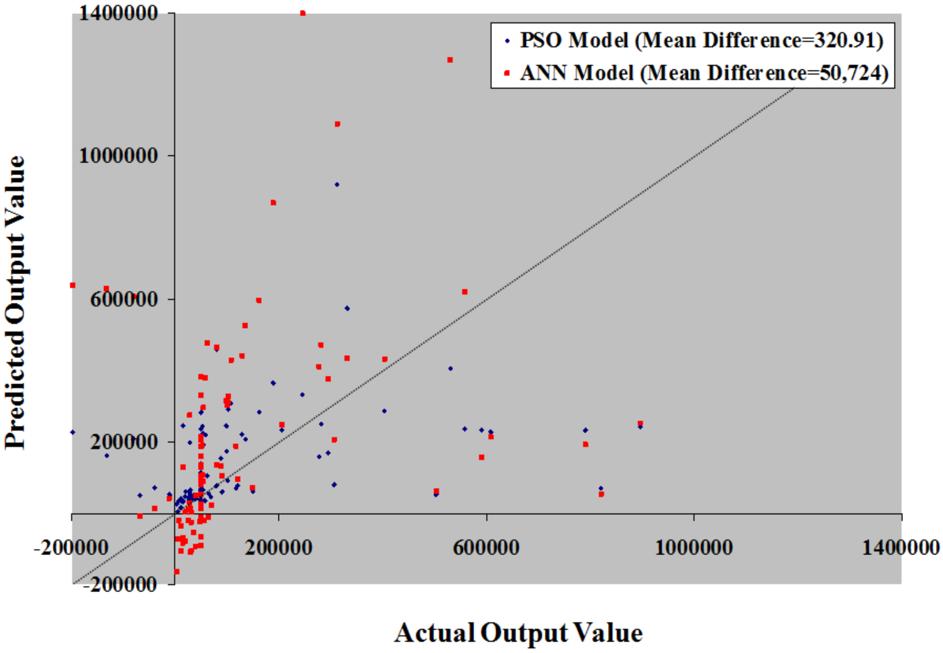

Figure 7. Correlation between actual and predicted output values for the validation data set

Both prediction models demonstrated positive bias, predicting higher contingencies on average than the actual values: $320 for the PSO model and $50,724 for the ANN model. However, the magnitude of the mean difference for the PSO model was negligible. .

5. ANALYSIS AND DISCUSSIONS

In order to determine the impact of an input variable on the output value, the “Importance of inputs” feature in the NeuroShell Predictor software was used. This value indicates the relative significance of each input variable in predicting the output value. The relative importance values range from 0 to 1 and they are normalized so that the sum of values for all input variables is 1. Figure 8 shows the relative importance of the four input variables for the PSO-based prediction model. The most significant input variables were found to be project amount and project duration.

The “Importance of inputs” parameter was used to direct pruning of the input variables from the model (a sort of a sensitivity analysis) measuring changes in prediction performance relative to a reduction in the number of input variables. The input variables with the lowest importance were the first to be removed. Figure 9 shows the performance of the four PSO-based prediction models relative to the number of input variables, for both the training and testing data sets. It was found that the prediction model with all four input variables had the best training performance with the average error increasing as the number of input variables from the model was increased. It can also be seen that the first two models (with 4 and 3 input variables respectively) had similarly good performances on the testing data set.

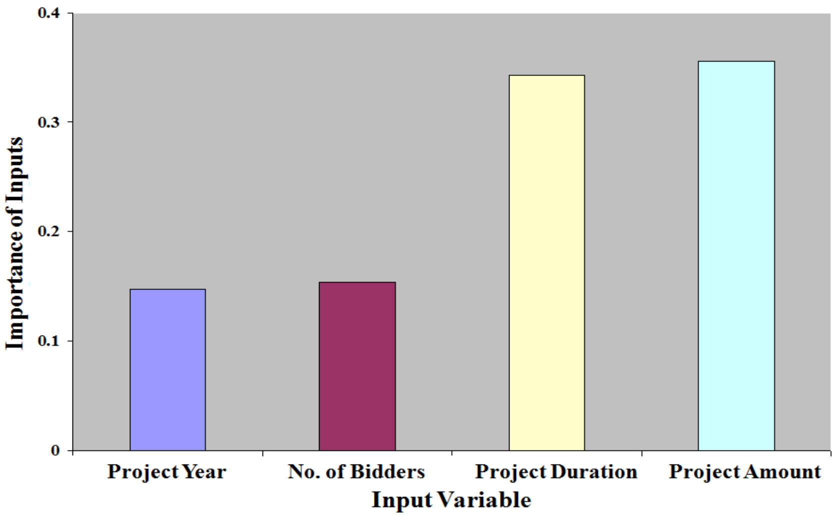

Figure 8. Importance of input variables on the PSO-based prediction model

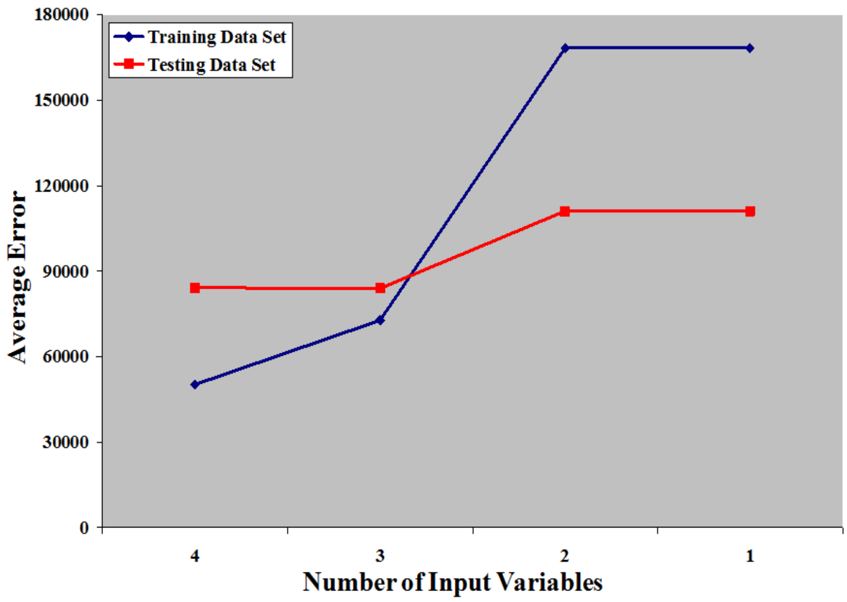

Figure 9. Sensitivity of average error on training and testing data sets to varying the number of input variables

A final analysis was made to determine whether the performance of the PSO-based model was consistent across the entire problem domain (the area defined by the range of values at the inputs). This was undertaken by plotting the errors for the testing data against each of the input variables, as shown in Figure 10. Visually, the PSO model performed relatively well across all values of the four input variables, although there is an apparent tendency to perform worse for lower values of the number of bidders and higher values of the project year. Alternatively, this might be explained by the fact that these worse performing regions occur where the testing patterns have the greatest density and thus where there is more opportunity to find extreme errors. In other words it may be attributable to bias resulting from a poor distribution of values across the problem domain. This indicates that more data are required for testing the PSO-based model in the low density data regions of the problem domain. Any regions then found to have inconsistently high testing errors may be improved by the use of additional training data at those locations.

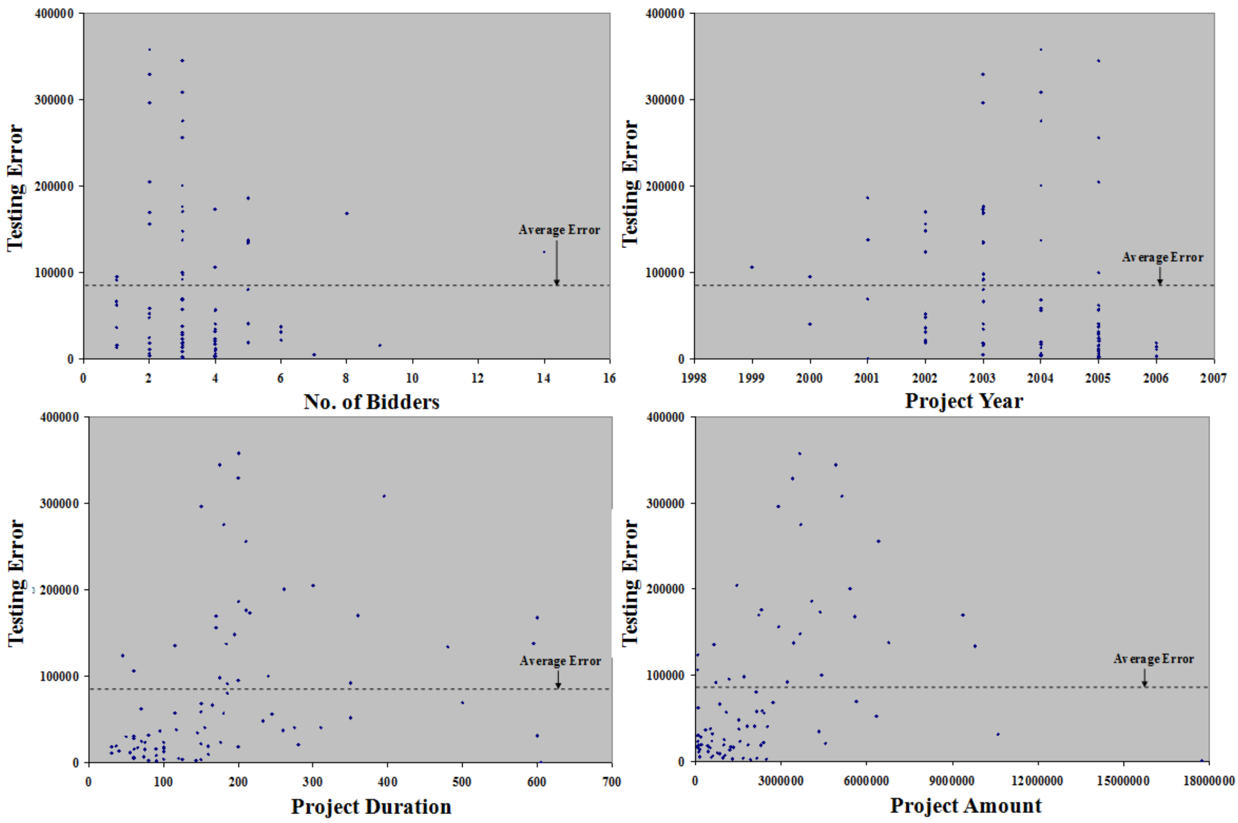

Figure 10. Evaluating testing error across problem domain for the PSO-based prediction model

6. CONCLUSIONS

The particle swarm optimization (PSO), a population-based evolutionary computation technique simulating social behaviors of bird flocking or fish schooling, has been applied to solve many optimization problems in civil engineering. PSO's main advantages are that it is relatively easy to implement, it has a relatively fast search process, and it can produce effective solutions. This study proposed a new intelligent computing approach to predict the owner's contingency amount based on the PSO methodology and demonstrated the viability of using the approach by comparing its performance with an existing ANN-based approach. The PSO-based prediction model employed in this study consisted of four input variables (the number of bidders, project year, project duration, and project amount) and one output variable (the owner's contingency amount). Using the "Importance of inputs" feature within the NeuroShell Predictor software to prune the input variables, it was found that project amount and project duration had the most impact on the owner's contingency amount and the project year could be removed without significantly impacting performance measured relative to the testing data.

The contributions of this study on the development of the proposed PSO-based prediction model include not only providing an alternative means to predict the cost contingency item, but also exploring the potential for applications of the PSO technique for prediction and forecasting problems in the construction area. Accurate predictions of contingency using this intelligent computing approach in the perspective of project owners and sponsors can be used to better manage project financial contingency requirements and allow for additional projects to be brought online at a faster pace under financially effective business plans.

Further studies for predicting more accurate contingency will address the following issues: identifying more input variables (such as number of concurrent contracts) and including them into the PSO-based prediction model, handling potential categorical input variables (such as project work type, project contract agreement type, and project geographical locations), comparing the performance of the model for an alternative form of output variable (such as the contingency rate), and comparing the performance of the PSO approach with other evolutionary computation techniques such as Genetic Algorithm (GA). Finally, although the PSO approach has provided valuable solutions to various optimization problems in the construction engineering field such as project scheduling, layout problems, and structural design, it also has potential for prediction and forecasting problems such as estimating project construction cost and duration, estimating productivity, and predicting performance.

7. REFERENCES

Ashuri, B. and Tavakolan, M. (2012). “Fuzzy enabled hybrid genetic algorithm-particle swarm optimization approach to solve TCRO problems in construction project planning.” *J. Construction Engineering Management,* 138(9).

Baccarini, D. (2004). “Accuracy in estimating project cost construction contingency: A statistical analysis.” *Proc., Int. Construction Research Conf. of the Royal Institution of Chartered Surveyors,* Headingley Cricket Club, Leeds. Australia.

Baccarini, D. and Love, P. E. D. (2014). “Statistical characteristics of cost contingency in water infrastructure projects.” *J. Construction Engineering Management,* 140(3).

Chen, Z., Cao, H., Ye, K., Zhu, H., and Li, S. (2015). “Improved particle swarm optimization-based form-finding method for suspension bridge installation analysis.” *J. Computing in Civil Engineering,* 29(3).

Dey, P., Tabucanon, M. T., and Ogunlana, S. O. (1994). “Planning for project control through risk analysis: A petroleum pipelaying project.” *International J. Project Management,* 12(1), 23-33.

Dimou, C. K., and Koumousis, V. K. (2009). “Reliability-based optimal design of truss structures using particle swarm optimization.” *J. Computing Civil Engineering,* 23(2), 100-109.

Eberhart, R. C., and Kennedy, J. (1995). “A new optimizer using particle swarm theory.” *Proc., 6th Int. Symposium on Micromachine and Human Science,* Nagoya, Japan, 39-43.

Eberhart, R. C., and Shi, Y. (1998). “Comparison between genetic algorithms and particle swarm optimization.” *Proc., 7th Annual Conf. on Evolutionary Programming: VII,* Springer, NY, 611-616.

Eberhart, R. C., and Shi, Y. (2001). “Tracking and optimizing dynamic systems with particle swarms.” *Proc., IEEE Congress on Evolutionary Computation,* New York, 94-97.

Florida Department of Transportation (FDOT) (2010), *Office of Construction Alternative Contracting,* Tallahassee, FL. <http://www.dot.state.fl.us/construction/AltContract/AltContract.shtm> (April 19, 2010).

Federal Highway Administration (FHWA) (2007). *Major project program cost estimating guidance,* Washington, DC.

Flood, I., and Issa, R. R. A (2010). “Empirical modeling methodologies for construction.” *J. Construction Engineering Management,* 136(1), 36-48.

Gopalakrishnan, K. (2010). “Neural network-swarm intelligence hybrid nonlinear optimization algorithm for pavement moduli back-calculation.” *J. Transportation Engineering, 136*(6), 528-48.

Gunhan, S., and Arditi, D. (2007). “Budgeting owner’s construction contingency.” *J. Construction Engineering Management,* 133(7), 492-497.

Karlsen, J., and Lereim, J. (2005). “Management of project contingency and allowance.” *Cost Engineering,* 47(9), 24-29.

Kennedy, J., and Eberhart, R. C. (1995). “Particle swarm optimization.” *Proc., IEEE Conf. on Neural Networks, IV*, NJ. Piscataway, NJ, 1942-1948.

Lhee, S. C., Issa, R. R. A., and Flood, I. (2012). “Prediction of financial contingency for asphalt resurfacing projects using artificial neural networks.” *J. Construction Engineering Management,* 138(1), 22-30.

Love, P. C., Sing, C., Carey, B., and Kim, J. T. (2015). "Estimating construction contingency: Accommodating the potential for cost overruns in road construction projects." *J. Infrastructure System,* 21(2).

Mak, S., and Picken, D. (2000). "Using risk analysis to determine construction project contingencies." *J. Construction Engineering Management,* 126(2), 130-136.

Molenaar, K. R. (2005). "Programmatic cost risk analysis for highway megaprojects." *J. Construction Engineering Management,* 131(3), 343-353.

Moselhi, O., Hegazy, T., and Fazio, P. (1993). "DBID: Analogy-based DSS for bidding in construction." *J. Construction Engineering Management,* 119(3), 466-479.

Oberlender, G. D., and Trost, S. M. (2001). "Predicting accuracy of early cost estimates based on estimate quality." *J. Construction Engineering Management,* 127(3), 173-182.

Popescu, C. M., Phaobunjong, K., and Ovararin, N. (2003). *Estimating building costs,* Marcel Dekker, New York.

Reddy, M. J., and Adarsh, S. (2010). "Overtopping probability constrained optimal design of composite channels using swarm intelligence technique." *J. Irrigation Drainage Engineering,* 136(8), 532-542.

Smith, G. R., and Bohn, C. M. (1999). "Small to medium contractor contingency and assumption of risk." *J. Construction Engineering Management,* 125(2), 101-108.

Thal, A. E., Cook, J. J., and White, E. D. (2010). "Estimation of cost contingency for Air Force construction projects." *J. Construction Engineering Management,* 136(11), 1181-1188.

Thompson, P. A., and Perry, J. G. (1992). *Engineering construction risks: A guide to project risk analysis and risk management,* Thomas Telford, London, U.K.

Touran, A. (2003). "Probabilistic model for cost contingency." *J. Construction Engineering Management,* 129(3), 280-284.

Wolpert, D. H., and Macready, W. G. (1997). "The no free lunch theorems for optimization." *IEEE Trans. Evol. Comput.,* 1(1), 67-82.

Yang, I. T. (2007). "Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis." *J. Construction Engineering Management,* 133(7), 498-505.

Yazdi, J. S., Kalantary, F., and Yazdi, H. S. (2012). "Calibration of soil model parameters using particle swarm optimization." *International J. Geomech.,* 12(3).

Zhang, H., Li, H., and Tam, C. M. (2006). "Permutation-based particle swarm optimization for resource-constrained project scheduling." *J. Computing in Civil Engineering,* 20(2), 141-149.

Zhang, H., and Wang, J. Y. (2008). "Particle swarm optimization for construction site unequal-area layout." *J. Construction Engineering Management,* 134(9), 739-748.