Lecture Notes in Computer Science, Vol. 4233, 2006, pp. 1101-1107

Prediction of Construction Litigation Outcome Using a Split-Step PSO Algorithm

Kwok-wing Chau

Department of Civil and Structural Engineering, Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong
cekwchau@polyu.edu.hk

Abstract. The nature of construction claims is highly complicated and the cost involved is high. It will be advantageous if the parties to a dispute may know with some certainty how the case would be resolved if it were taken to court. The recent advancements in artificial neural networks may render a cost-effective technique to help to predict the outcome of construction claims, on the basis of characteristics of cases and the corresponding past court decisions. In this paper, a split-step particle swarm optimization (PSO) model is applied to train perceptrons in order to predict the outcome of construction claims in Hong Kong. It combines the advantages of global search capability of PSO algorithm in the first step and the local convergence of back-propagation algorithm in the second step. It is shown that, through a real application case, its performance is much better than the benchmark backward propagation algorithm and the conventional PSO algorithm.

1 Introduction

The nature of construction activities is varying and dynamic, which can be evidenced by the fact that no two sites are exactly the same. Thus the preparation of the construction contract can be recognized as the formulation of risk allocation amongst the involving parties: the client, the contractor, and the engineer. The risks involved include the time of completion, the final cost, the quality of the works, inflation, inclement weather, shortage of materials, shortage of plants, labor problems, unforeseen ground conditions, site instructions, variation orders, client-initiated changes, engineer-initiated changes, errors and omissions in drawings, mistakes in specifications, defects in works, accidents, supplier delivery failure, delay of schedule by subcontractor, poor workmanship, delayed payment, changes in regulations, third-party interference, professional negligence, and so on.

Before the actual construction process, the involving parties will attempt to sort out the conditions for claims and disputes through the contract documents. However, since a project usually involves thousands of separate pieces of work items to be integrated together to constitute a complete functioning structure, the potential for honest misunderstanding is extremely high. The legislation now in force requires that any disputes incurred have to be resolve successively by mediation, arbitration, and the courts [1].

By its very nature, the construction industry is prone to litigation since claims are normally affected by a large number of complex and interrelated factors. However, the consequence of any disagreements between the client and the contractor may be far reaching. It may lead to damage to the reputation of both sides, as well as inefficient use of resources and higher costs for both parties through settlement. The litigation process is usually very expensive since it involves specialized and complex issues. Thus, it is the interest of all the involving parties to minimize or even avoid the likelihood of litigation through conscientious management procedure and concerted effort. It is highly desirable for the parties to a dispute to know with some certainty how the case would be resolved if it were taken to court. This would effectively help to significantly reduce the number of disputes that would need to be settled by the much more expensive litigation process.

Recently, soft computing (SC) techniques have been gradually becoming a trend. The characteristics of these data-driven approaches include built-in dynamism, data-error tolerance, no need to have exogenous input and so on. Amongst others, artificial neural networks (ANN), in particular the feed forward back-propagation (BP) perceptrons, have been widely applied in different fields [2-6]. The use of ANN can be a cost-effective technique to help to predict the outcome of construction claims, on the basis of characteristics of cases and the corresponding past court decisions. It can be used to identify the hidden relationships among various interrelated factors and to mimic decisions that were made by the court. However, slow training convergence speed and easy entrapment in a local minimum are inherent drawbacks of the commonly used BP algorithm [7]. Swarm intelligence is another recent SC technique that is developing quickly [8]. These SC techniques have been applied successfully to different areas [9-12].

This paper presents a split-step PSO algorithm which is employed to train multi-layer perceptrons for prediction of the outcome of construction litigation in Hong Kong. It is believed that, by combining the two algorithms, the advantages of global search capability of PSO algorithm in the first step and local convergence of BP algorithm in the second step can be fully utilized to furnish promising results. This paper contributes to the verification of this new algorithm to real prototype application. It can be extended and applied to other areas as well.

2 Split-Step PSO Algorithm

The combination of two different SC techniques could enhance the performance through fully utilization of the strengths of each technique. In this algorithm, the training process is divided into two stages. Initially the perceptron is trained with the PSO algorithm for a predetermined generation number to exploit the global search ability for near-optimal weight matrix. Then, after this stage, the perceptron is trained with the BP algorithm to fine tune the local search. This might be able to avoid the drawback of either entrapment in local minima in BP algorithm or longer time consumption in global search of PSO algorithm.

2.1 PSO Algorithm

When PSO algorithm is initially proposed, it is considered a tool for modeling social behavior and for optimization of difficult numerical solutions [8,13]. This computational intelligence technique is intimately related to evolutionary algorithms and is an optimization paradigm that mimics the ability of human societies to process knowledge [14]. Its principle is based on the assumption that potential solutions will be flown through hyperspace with acceleration towards more optimum solutions. PSO is a populated search method for optimization of continuous nonlinear functions resembling the biological movement in a fish school or bird flock. Each particle adjusts its flying according to the flying experiences of both itself and its companions. During the process, the coordinates in hyperspace associated with its previous best fitness solution and the overall best value attained so far by other particles within the group are kept track and recorded in the memory.

One of the more significant advantages is its relatively simple coding and hence low computational cost. One of the similarities between PSO and a genetic algorithm is the fitness concept and the random population initialization. However, the evolution of generations of a population of these individuals in such a system is by cooperation and competition among the individuals themselves. The population is responding to the quality factors of the previous best individual values and the previous best group values. The allocation of responses between the individual and group values ensures a diversity of response. The principle of stability is adhered to since the population changes its state if and only if the best group value changes. It is adaptive corresponding to the change of the best group value. The capability of stochastic PSO algorithm, in determining the global optimum with high probability and fast convergence rate, has been demonstrated in other cases [13-14].

2.2 Training of Three-Layered Perceptrons

PSO can be readily adopted to train the multi-layer perceptrons as an optimization technique. In the following section, a three-layered preceptron is considered, although the same principle still holds for other number of layers. $W^{[1]}$ and $W^{[2]}$ represent the connection weight matrix between the input layer and the hidden layer, and that between the hidden layer and the output layer, respectively. During training of the preceptron, the i-th particle is denoted by $W_i = \{W^{[1]}, W^{[2]}\}$ whilst the velocity of particle i is denoted by V_i. The position representing the previous best fitness value of any particle is denoted by P_i whilst the best matrix among all the particles in the population is recorded as P_b. Let m and n represent the index of matrix row and column, respectively, the following equation represents the computation of the new velocity of the particle based on its previous velocity and the distances of its current position from the best experiences both in its own and as a group.

$$\begin{aligned} V_i^{[j]}(m,n) = V_i^{[j]}(m,n) + r\alpha[P_i^{[j]}(m,n) - W_i^{[j]}(m,n)] \\ + s\beta[P_b^{[j]}(m,n) - W_i^{[j]}(m,n)] \end{aligned} \quad (1)$$

where j = 1, 2; m = 1, …, M_j; n= 1, …, N_j; M_j and N_j are the row and column sizes of the matrices W, P, and V; r and s are positive constants; α and β are random numbers in the range from 0 to 1. In the context of social behavior, the cognition part $r\alpha[P_i^{[j]}(m,n) - W_i^{[j]}(m,n)]$ denotes the private thinking of the particle itself whilst the social part $s\beta[P_b^{[j]}(m,n) - W_i^{[j]}(m,n)]$ represents the collaboration among the particles as a group. The new position is then determined based on the new velocity as follows:

$$W_i^{[j]} = W_i^{[j]} + V_i^{[j]} \tag{2}$$

The fitness of the i-th particle is determined in term of an output mean squared error of the neural networks as follows:

$$f(W_i) = \frac{1}{S}\sum_{k=1}^{S}\left[\sum_{l=1}^{O}\{t_{kl} - p_{kl}(W_i)\}^2\right] \tag{3}$$

where f is the fitness value, t_{kl} is the target output; p_{kl} is the predicted output based on W_i; S is the number of training set samples; and, O is the number of output neurons.

3 Application to Construction Litigation

In this study, the system is applied to predict the outcome of construction claims in Hong Kong. The existing data from 1991 to 2000 are pre-processed initially and organized case by case in order to correlate the relationship between the dispute characteristics and court decisions. Through a sensitivity analysis, 13 case elements that seem relevant in courts' decisions, which are namely, type of contract, contract value, parties involved, type of plaintiff, type of defendant, resolution technique involved, legal interpretation of contract documents, misrepresentation of site, radical changes in scope, directed changes, constructive changes, liquidated damages involved, and late payment, are identified.

As far as possible, the 13 case elements are expressed in binary format; for example, the input element 'liquidated damages involved' receives a 1 if the claim involves liquidated damages or a 0 if it does not. However, some elements are defined by several alternatives; for example, 'type of contract' could be remeasurement contract, lump sum contract, or design and build contract. These elements with alternative answers are split into separate input elements, one for each alternative. Each alternative is represented in a binary format, such as 1 for remeasurement contract and 0 for the others if the type of contract is not remeasurement. In that case, only one of these input elements will have a 1 value and all the others will have a 0 value. In this way, the 13 elements are converted into an input layer of 30 neurons, all expressed in binary format. The court decisions are also organized in an output layer of 6 neurons expressed in binary format corresponding to the 6 elements: client, contractor, engineer, sub-contractor, supplier, and other third parties. Table 1 shows examples of the input neurons for cases with different types of contract.

Table 1. Examples of the input neurons for cases with different types of contract

Input neuron	Cases		
	Remeasurement	Lump sum	Design and build
Type of contract - remeasurement	1	0	0
Type of contract - lump sum	0	1	0
Type of contract – design and build	0	0	1

In this case, 1105 sets of construction-related cases are employed, of which 550 from years 1991 to 1995 are used for training, 275 from years 1996 to 1997 are used for testing, and 280 from years 1998 to 2000 are used to validate the network results with the observations. In the PSO-based perceptron, the number of population is set to be 40 whilst the maximum and minimum velocity values are 0.25 and -0.25 respectively. In forming the data series for training and validation, a balanced distribution of cases is ensured. In order to determine the best architecture, a sensitivity analysis is undertaken to vary in the number of hidden layers and number of hidden neurons. After a lot of numerical experiments, the final perceptron is determined. Table 2 shows the parameters for the best architecture.

Table 2. Parameters for the best architecture

	Parameter
No. of hidden layer	3
No. of neuron in input layer	30
No. of neuron in hidden layer	15
No. of neuron in output layer	6

Table 3. Comparison of prediction results for various perceptrons

Algorithm	Training		Validation	
	Coefficient of correlation	Prediction rate	Coefficient of correlation	Prediction rate
BP-based	0.956	0.69	0.953	0.67
PSO-based	0.987	0.81	0.984	0.80
Split-step	0.988	0.83	0.985	0.82

Table 4. Steady-state fitness evaluation times during training for various perceptrons

Algorithm	Steady-state fitness valuation time
BP-based	22,400
PSO-based	8,300
Split-step	7,900

4 Analysis and Discussions

In evaluating the performance of the split-step multi-layer ANN, a comparison is made with several commonly used existing methods, i.e., the benchmarking standard BP-based network and a PSO-based network. A fair and common initial ground is ensured for comparison purpose as far as possible. The training process of the BP-based perceptron commences from the best initial population of the corresponding PSO-based perceptron or split-step network. Table 3 shows comparisons of the results of network for various perceptrons. It can be observed that the split-step algorithm performs the best in terms of prediction accuracy. It is noted that testing cases of the split-step PSO-based network are able to give a successful prediction rate higher than 80%, which is much higher than by pure chance.

Table 4 shows the steady-state fitness evaluation times during training for various perceptrons. The fitness evaluation time here for the PSO-based perceptron is equal to the product of the population with the number of generations. It can be observed that the split-step perceptron exhibits much faster convergence than those by the BP-based perceptron and the PSO-based network. It is, of course, recognized that there are limitations in the assumptions used in this study. Other factors that may have certain bearing such as cultural, psychological, social, environmental, and political factors have not been considered here.

5 Conclusions

This paper presents the application of a perceptron based on a split-step PSO algorithm for prediction of outcomes of construction litigation on the basis of the characteristics of the individual dispute and the corresponding past court decisions. It is believed that, if the involving parties to a construction dispute become aware with some certainty how the case would be resolved if it were taken to court, the number of disputes could be reduced significantly. It is shown that the split-step PSO-based perceptron performs much better than the other commonly used optimization techniques in prediction of outcomes of construction litigation. The rate of prediction for the network finally adopted in this study is higher than 80%, which is much higher than pure chance. It can be used as a good prediction tool for the parties in dispute.

References

1. Chau, K.W.: Resolving Construction Disputes by Mediation: Hong Kong Experience. Journal of Management in Engineering, ASCE **8(4)** (1992) 384-393
2. Chau, K.W.: A Review on the Integration of Artificial Intelligence into Coastal Modeling. Journal of Environmental Management **80(1)** (2006) 47-57
3. Chau, K.W., Cheng, C.T.: Real-Time Prediction of Water Stage with Artificial Neural Network Approach. Lecture Notes in Artificial Intelligence **2557** (2002) 715-715
4. Cheng, C.T., Chau, K.W., Sun, Y.G., Lin, J.Y.: Long-term Prediction of Discharges in Manwan Reservoir using Artificial Neural Network Models. Lecture Notes in Computer Science **3498** (2005) 1040-1045
5. Cheng, C.T., Lin, J.Y., Sun, Y.G., Chau, K.W.: Long-Term Prediction of Discharges in Manwan Hydropower using Adaptive-Network-Based Fuzzy Inference Systems Models. Lecture Notes in Computer Science **3612** (2005) 1152-1161
6. Wu, C.L. Chau, K.W.: Evaluation of Several Algorithms in Forecasting Flood. Lecture Notes in Artificial Intelligence **4031** (2006) 111-116
7. Rumelhart, D.E., Widrow, B., Lehr, M.A.: The Basic Ideas in Neural Networks. Communications of the ACM **37(3)** (1994) 87-92
8. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth (1995) 1942-1948
9. Chau, K.W.: River Stage Forecasting with Particle Swarm Optimization. Lecture Notes in Artificial Intelligence **3029** (2004) 1166-1173
10. Chau, K.W.: Rainfall-Runoff Correlation with Particle Swarm Optimization Algorithm. Lecture Notes in Computer Science **3174** (2004) 970-975
11. Chau, K.W.: Predicting Construction Litigation Outcome using Particle Swarm Optimization. Lecture Notes in Artificial Intelligence **3533** (2005) 571-578
12. Arditi, D., Oksay, F.E., Tokdemir, O.B.: Predicting the Outcome of Construction Litigation Using Neural Networks. Computer-Aided Civil and Infrastructure Engineering **13(2)** (1998) 75-81
13. Kennedy, J.: The Particle Swarm: Social Adaptation of Knowledge. Proceedings of the 1997 International Conference on Evolutionary Computation. Indianapolis (1997) 303-308
14. Clerc, M., Kennedy, J.: The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation **6(1)** (2002) 58-73