

# Reconstruction of an Expert's Decision Making Expertise in Concrete Dispatching by Machine Learning

Mojtaba Maghrebi<sup>1</sup>, Claude Sammut<sup>2</sup> and Travis Waller<sup>1,3</sup>

1. School of Civil and Environmental Engineering, The University of New South Wales (UNSW), Kensington NSW 2052, Australia

2. School of Computer Science Engineering, The University of New South Wales (UNSW), Kensington NSW 2052, Australia

3. National Information and Communications Technology Australia (NICTA), Sydney 20000, Australia

**Abstract:** Finding the optimum solution for dispatching in concrete delivery is computationally intractable because it is a NP-hard (non-deterministic polynomial-time hard) problem. Heuristic methods are required to obtain satisfactory solutions. Inefficiencies in mathematical modeling still make concrete dispatching difficult to solve. In reality, complex dispatching systems are mostly handled by human experts, who are able to manage the assigned tasks well. However, the high dependency on human expertise is a considerable challenge for RMC (ready mixed concrete) companies. In this paper, a logical reconstruction of an expert's decision making is achieved by two machine learning techniques: decision tree and rule induction. This paper focuses on the expert dispatcher's prioritization of customer orders. The proposed method has been tested on a simulation model consisting of a batch plant and three customers per day. The scenarios generated by the simulation model were given to a dispatch manager who was asked to prioritize the customers in each day. The scenarios and the decisions were then input to the machine learning programs, which created generalizations of the expert's decisions. Both decision trees and rules approach 80% accuracy in reproducing the human performance.

**Key words:** Machine learning, logic reconstruction, experts' behavior.

## 1. Introduction

Concrete is the single most widely used material in the world [1]. Due to technical requirements, usually concrete must be mixed in a batch plant and hauled to the construction site. Typically, the capacity of a RMC (ready mixed concrete) company is limited by the size of its fleet, hence the importance of truck dispatching for RMCs. Optimization is the first technique that comes to mind to solve RMC dispatching problems. There are many examples in the literature that have been proposed to solve this problem [2-7], however, the RMC dispatching problem is NP-hard, which means that there is no method that can solve this problem in polynomial time [2, 4, 6, 7]. Thus, obtaining optimum decisions is computationally intractable. In the literature, approximate, heuristic methods have been widely used by researchers to

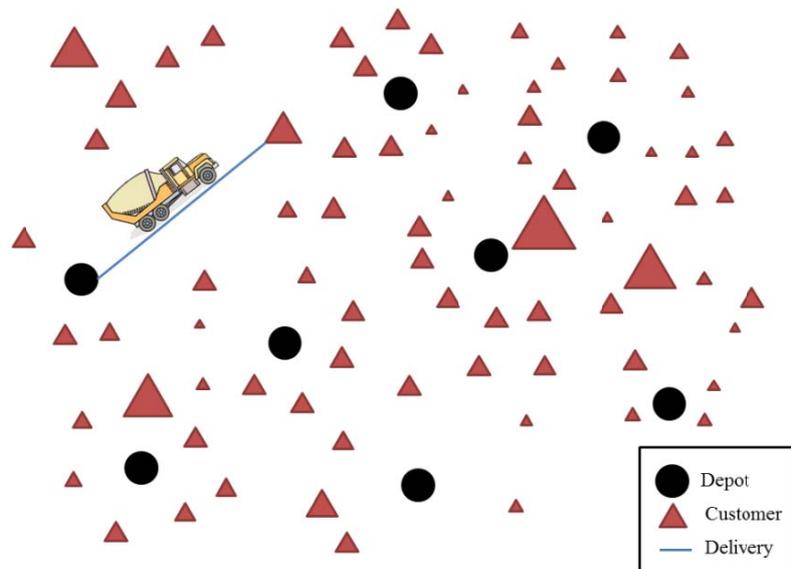
achieve an acceptable result [2-4, 6-10]. In heuristic methods, the mathematical model, which is built for optimization is used. Even though the results achieved by these techniques are acceptable, there are still many problems in mathematical modelling that threaten the reliability of the techniques. To investigate this issue thoroughly, an RMC mathematical model that was introduced by Ref. [2], and used by other scholars such as Ref. [11], is described here.

RMC dispatching is represented by a network, as Fig. 1. In this network, each customer and depot is allocated a node. For each delivery, a connection is needed between a depot and a customer. The amount of concrete ordered by different customers is distinct. The number of required deliveries is calculated for each project, based on the ordered amounts. The size of a triangle in Fig. 1 reflects the amount of purchased concrete.

Although the mathematical modelling techniques are powerful tools, there are many problems, especially

---

**Corresponding author:** Mojtaba Maghrebi, Ph.D. candidate, research fields: construction management and machine learning. E-mail: maghrebi@unsw.edu.au.



**Fig. 1** A dispatching network.

when they are used to model an uncertain dispatching system such as concrete delivery. In the next section, we will cover two of the most important problems of mathematic modelling. In the subsequent section, the inefficiencies of mathematical modelling are discussed along with the reasoning behind why such systems can be handled by human experts. The main aim of this paper is to extract knowledge from an expert's decisions by machine learning and build a platform that will be able to make dispatching decisions automatically.

## 2. Insight into the Problem

In this section, the inefficiencies of mathematical models in coping with real situations are discussed.

### 2.1 Large Number of Variables

Mathematical modelling consists of an objective function that reflects the desired goals and constraints of the problem. All conditions must be casted into constraints, even the trivial issues. Therefore, the first problem in building the constraint model is the large number of variables that are needed to express the constraints. For example, in RMC dispatching with nine depots and around 270 deliveries with 70 trucks,

the model requires more than 67 million variables. Regardless of the technique that is selected to solve this problem, the number of variables is a challenge.

### 2.2 Dealing with Uncertain and Dynamic Data

RMC dispatching systems are very uncertain and dynamic, since it is expected that problem conditions be changed at any time [2, 6, 10, 11]. Any change in data requires an adjustment in the model. For example, a minor change in the system, such as a delay of a truck, needs a minor adjustment in model and a major change, such as a truck breakdown, needs a major adjustment in the model. So, RMC data are dynamically changing, so handling requires constant review and dynamic adjustment during the operation. If the amount change is significant then the adjusted model must be reruned. This take times, despite the fact that a dispatching system needs on-time decisions.

Due to the discussed issues and in despite of much research in this field, RMC dispatching suffers from a lack of practical solutions [8, 12]. Consequently, in practice, RMC dispatches are handled by experts and who are able to manage the tasks that are assigned to them efficiently. In this paper, the authors attempt to extract the knowledge from the decisions of an expert

dispatcher. Many types of decisions are made in a day, however, the authors focus on customer prioritization.

2.3 Role of Experts in RMC Dispatching

RMC dispatchers are expected to satisfy customer orders, so they must find ways to deliver the ordered concrete during the day with minimum delay and cancelations. In reality, there are many parameters that affect the decisions that an expert takes (Fig. 2).

It should be emphasized that a dispatcher must simultaneously consider all of these parameters in reaching a profitable decision. Moreover, prioritizing customers is one of the most important decisions that the dispatchers make to achieve a profit. Since the performance of expert dispatchers is acceptable to RMCs and the nature of their jobs does not change significantly, it should be possible to construct rules that model their decision making.

3. Methodology

Machine learning is used to extract the logic of an expert’s decisions from traces of his behavior. In the present experiments, decision-trees and rule-based

representations were selected for their simplicity and readability.

Decision tree is have been widely used in machine learning and data mining [13]. As the authors use the Weka machine learning toolkit, the J48 algorithms was selected. This is a reimplementaion of Quinlan’s C4.5 [14]. It uses a “divide-and-conquer” approach to build a tree structure that consists of internal nodes, branches and leaf nodes. An internal node tests a particular attribute and based on the possible values for the attribute, a branch is selected. A leaf node represents a class, so when an instance reaches a leaf node, the leaf’s class is assigned to the instance. Decision trees can handle noisy data by pruning parts of the tree that may be due to over fitting [14].

The part algorithm [15] creates a set of rules, from partial decision trees with the aim of avoiding the two pass method of C4.5 for creating an optimal set of rules [16].

4. Case Study

To test the proposed method, a simulation model was built that consists of a batch plant and three

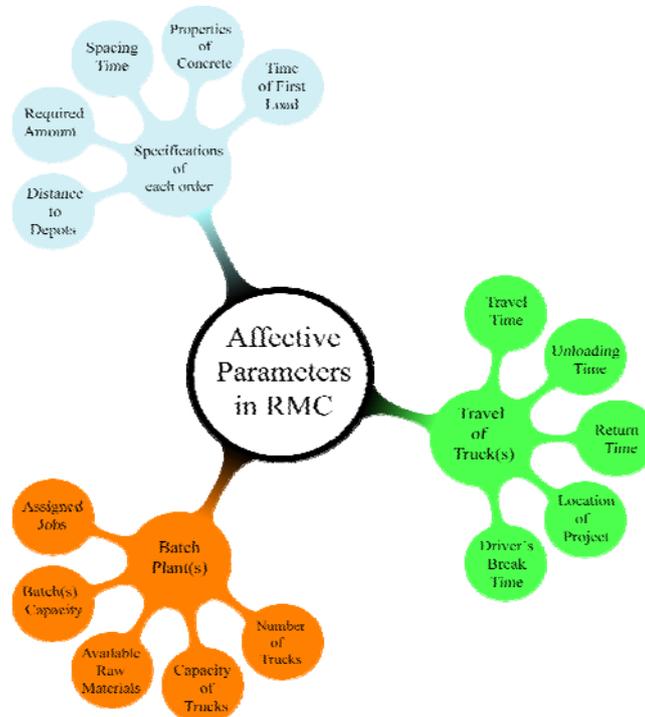


Fig. 2 Summary of affective parameters in RMC (ready mixed concrete).

customers. The simulation model randomly generated the customer's attributes for 200 days. The generated values for each attribute are within realistic ranges. A dispatch manager was asked to look at the available data for all customers and make a decision about the priority level of each customer. The training sets for machine learning were constructed from traces of the dispatcher's decision making behaviour. A training example includes following parameters:

- (1) day of delivery (Saturday-Friday);
- (2) the amount of ordered concrete for each project ( $m^3$ );
- (3) travel time (min);
- (4) spacing time (min);
- (5) location of the projects (name of suburbs);
- (6) the priority of customers.

Optimization techniques only accept numerical values and in the objective function all the units must be the same. For machine learning, the training sets include a combination of nominal and numerical variables. Also the selected machine learning techniques are capable of accepting numerical attributes with different units.

The number of projects per day in a simulation model can be varied, however, a simple RMC case was selected to help the researchers understand the behaviour of the machine learning techniques and to give them some insight into the structure of the database. In each instance, the dispatcher was asked to simultaneously consider the locations of projects, the day of delivery in the week and the specifications of all orders in order to make a decision about the priority of projects.

To make the simulation realistic, an actual metropolitan area consisting of seven suburbs was selected. In this area, there is a batch plant that supplies concrete for all seven suburbs. The data by the simulation generated was sent to the dispatch manager of that batch plant for him to prioritize the projects in each day. 200 instances were prioritized by the dispatcher in two stages with each stage involving 100

instances.

## 5. Results and Discussion

In building the training sets, a class must be assigned to each instance. The machine learning algorithms are expected to find a relation between the attribute values of an instance and the assigned classes. The expert prioritized the customers by providing a string for each set. For example, if the response of the expert for an instance is "3-2-1" this means that Customer#3 has the highest priority which is followed by Customer#2 and then Customer#1. On the other hand, the number of training sets is not enough for considering all possible classes in a learning process. So for each customer a separate learning process was taken into the account. So, the assigned level of priority was defined as a class. For example with reference to the above case the decision class for Customer#1 is Third, Customer#2 is Second and Customer#3 is First. Consequently, the number of classes was dropped from 6 to 3. It provides more training sets for each class and more accurate results. However, the training process was taken separately for each customer. The achieved results for all customers are very close to others, so for avoiding redundancy only the results for customer#1 are discussed.

In the training process, the same dataset was used in the two selected algorithms, and the 10 folds cross-validation was selected for evaluating the picked algorithms. It means that the dataset was divided into 10 folds with around 90% of each fold used for training and the remaining 10% of data was used for testing. Then in each fold the performance of algorithms was calculated for 11 performance metrics, which is further explained below. Table 1 shows the average of achieved results from 10 folds for each feature. The performance metrics used for comparing the performance of algorithms are: ACC (accuracy), FSC (F-score), APR (average precision), precision/recall BEP (break-evenpoint), RMSE (root mean squared error), CVMS (cross-validation mean sensitivity),

MSP (mean specificity), AUC (area under the ROC), SAR = (ACC + AUC + (1 - RMSE)) / 3, model building time and model testing time.

In terms of comparison, the most important feature is ACC. It reflects the ability of each algorithm in identifying the correct decisions made by a classifier. According to the achieved results, in terms of accuracy the performance of J48 with a trivial difference is better than part. Instances with algorithms that are slightly different due to randomness issues can not identify which algorithm outperforms others just based on accuracy [17-19]. Moreover, the best algorithm in one feature is definitely not the best in other performance metrics. In similar cases, the significance test is recommended. The *t*-test is a common significance test for machine learning techniques, which is used for

comparing machine learning algorithm in one domain [13, 20]. The *t*-test investigates the meaningful difference between a pair of algorithms. There are two hypotheses in a *t*-test, H0 for equality and H1 for inequality at an  $\alpha$  level of significance. The *P* value is calculated through a *t*-test and if it is lower than  $\alpha$  then H1 will be accepted; otherwise the H0 is true which reflects that there is no significant difference between two algorithms. 10 fold cross-validations are used to calculate the error rate for each fold, and then the 10 calculated error values for each algorithm were used in a paired *t*-test. The derived result of the paired *t*-test shows that the *P* value (0.8088) is greater than 0.01, so can not be proven that there is a significant difference between the two algorithms. It means that from a machine learning perspective a 99% level of

Table 1 Results of selected algorithms.

	ACC	FSC	APR	PRC	RMSE	CVMS	MSP	AUC	SAR	Building time (s)	Testing time (s)
J48	78.921	0.785	0.798	0.798	0.356	0.795	0.104	0.848	0.761	0.0015	0.0000
PART	79.395	0.785	0.797	0.797	0.351	0.802	0.112	0.876	0.773	0.0109	0.0000

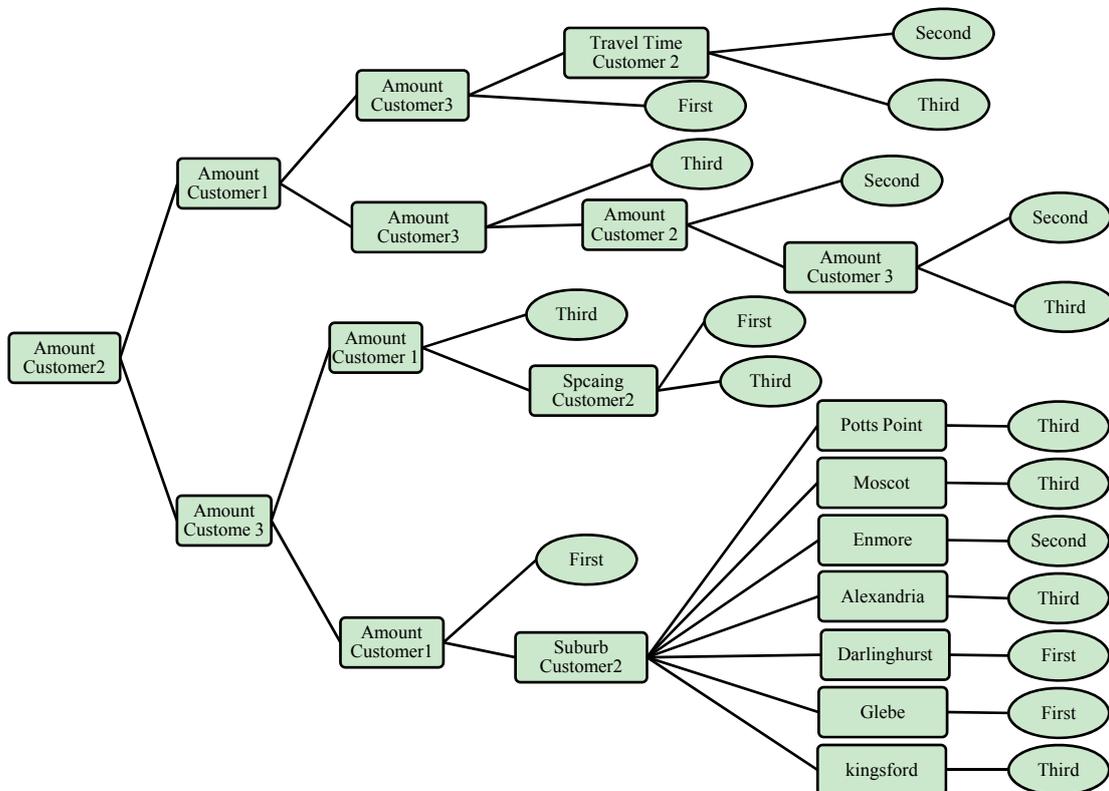
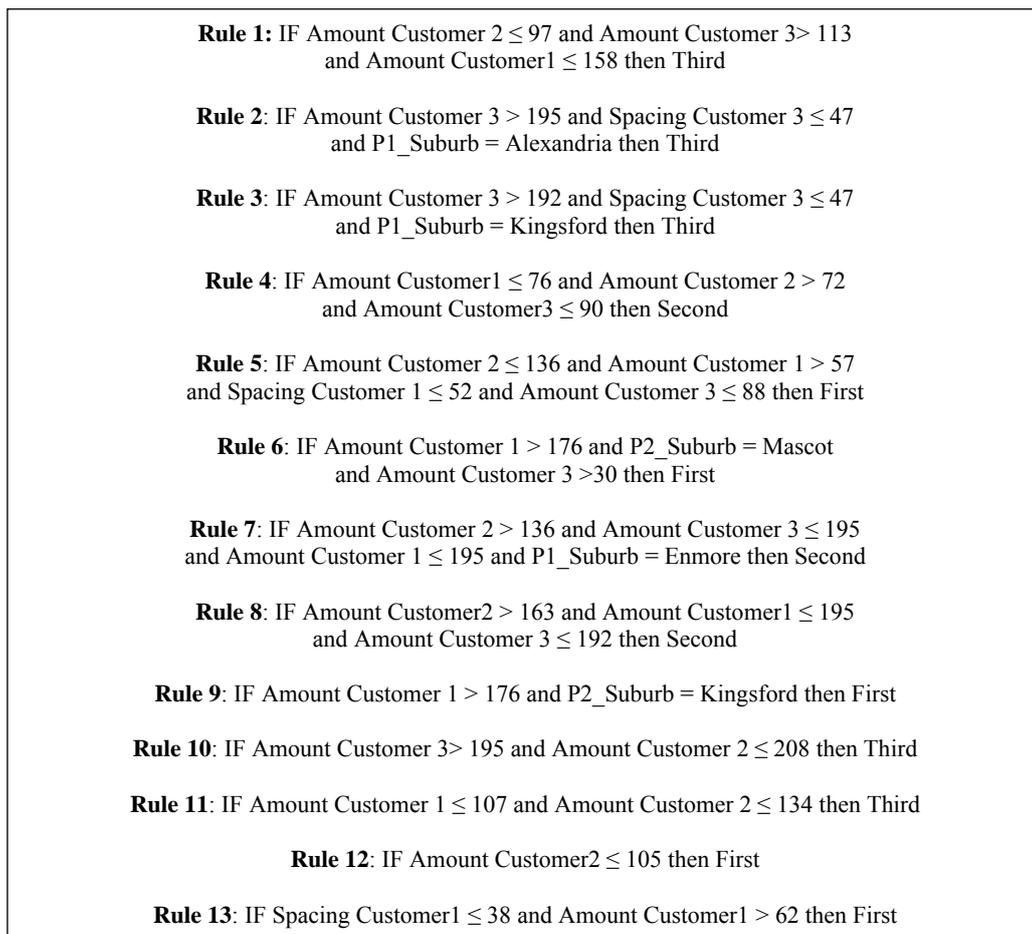


Fig. 3 Decision tree produced by J48.



**Fig. 4** Derived rules by PART.

confidence implies that the performances of both of the selected algorithms are the same. Consequently, the authors accept the performance of both algorithms and the built structures by J48 and PART which are respectively illustrated in Figs. 3 and 4.

At first glance at the results, one can realize that the most important parameter that the expert dispatcher considered is the amount of ordered concrete as shown in Figs. 3 and 4 as under the parameter name “Amount Customer  $i$ ”. The results point out that the higher priorities were mostly given to larger orders. The next interesting point that is observed from the results is that the priority of a customer is assessed not just according to its own features but also by considering the features of other customers. This point supports the real situations/real nature of the experts’ job and reflects how the selected algorithm solved the problem.

Moreover, timing attributes and especially the travel time between the batch plant and the customer were expected to be affective parameters in the decision making process. However, the results show that these parameters are not very important for dispatchers and that the information about the location of customers plays a key role in the decisions of an expert. In other words, the dispatcher mostly took into account the location of customers with hardly any consideration to the timing parameters. The expert dispatcher practically realized the distances between the customers and the batch plant and the other related information such as travel time in making decisions. Moreover, the other important feature that is embedded in location is the traffic pattern. Some suburbs are located in crowded areas where traffic is very dense in rush hours. Therefore, information about the location

of a customer provides a lot of information to an expert. In the conducted case study, sometimes the dispatcher preferred to postpone a large order by giving a lower level of priority in order to supply the concrete faster and more efficiently. In summary, the structures of the built decision tree and rules are totally different, however, both algorithms depict a very similar logic in the human's decisions.

## 6. Conclusions

In this paper, the extraction of the logic from an expert's decision was conducted in order to build a system that will be able to automatically handle the concrete dispatching. To test the proposed method, a simulation model with a batch plant and three projects in each day was built. The simulated scenarios were given to an expert dispatcher who was asked to prioritize the customers. Then based on the simulation data and the expert's decisions a training process was conducted on two machine learning algorithms; decision tree (J48) and rules-based (PART). In terms of accuracy both algorithms obtained similar results. Lastly, though the structures of the decision tree and rules based are totally different, the derived logic from both algorithms was very similar. It shows the ways that experts deal with a complex system such as concrete delivery problem can be discovered by machine learning techniques. In the future of research, the logic extract from large scale RMC dispatching will be conducted by generalizing problem in machine learning techniques.

## Acknowledgments

NICTA is funded by the Australian Department of Communications and the Australian Research Council through the ICT Centre of Excellence program.

## References

- [1] P.C. Aitcin, Cements of yesterday and today: Concrete of tomorrow, *Cement and Concrete Research* 30 (2000) 1349-1359.
- [2] L. Asbach, U. Dorndorf, E. Pesch, Analysis, modeling and solution of the concrete delivery problem, *European Journal of Operational Research* 193 (2009) 820-835.
- [3] P.C. Lin, J. Wang, S.H. Huang, Y.T. Wang, Dispatching ready mixed concrete trucks under demand postponement and weight limit regulation, *Automation in Construction* 19 (2010) 798-807.
- [4] S. Yan, W. Lai, An optimal scheduling model for ready mixed concrete supply with overtime considerations, *Automation in Construction* 16 (2007) 734-744.
- [5] C.W. Feng, T.M. Cheng, H.T. Wu, Optimizing the schedule of dispatching RMC trucks through genetic algorithms, *Automation in Construction* 13 (2004) 327-340.
- [6] S. Yan, W. Lai, M. Chen, Production scheduling and truck dispatching of ready mixed concrete, *Transportation Research Part E: Logistics and Transportation Review* 44 (2008) 164-179.
- [7] M. Maghrebi, T. Waller, C. Sammut, Scheduling concrete delivery problems by a robust meta heuristic method, in: *Sevens European Symposium on Computer Modeling and Simulation (EMS)*, UK, 2013.
- [8] D. Naso, M. Surico, B. Turchiano, U. Kaymak, Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete, *European Journal of Operational Research* 177 (2007) 2069-2099.
- [9] V. Schmid, K.F. Doerner, R.F. Hartl, M.W.P. Savelsbergh, W. Stoecher, A hybrid solution approach for ready-mixed concrete delivery, *Transportation Science* 43 (2009) 70-85.
- [10] V. Schmid, K.F. Doerner, R.F. Hartl, J.J. Salazar-González, Hybridization of very large neighborhood search for ready-mixed concrete delivery problems, *Computers & Operations Research, Operations Research* 37 (2010) 559-574.
- [11] Z. Yang, L. Menglei, L. Zhenyuan, Vehicle scheduling and dispatching of ready mixed concrete, in: *Fourth International Workshop on Advanced Computational Intelligence (IWACI)*, China, 2011, pp. 465-472.
- [12] C.W. Feng, H.T. Wu, Integrating FMGA and CYCLONE to optimize the schedule of dispatching RMC trucks, *Automation in Construction* 15 (2006) 186-199.
- [13] I.H. Witten, E. Frank, *Data Mining, Practical Machine Learning Tolls and Techniques*, Elsevier, New York, 2005.
- [14] R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, 1992.
- [15] E. Frank, I.H. Witten, Generating accurate rule sets without global optimization, in: *Proceedings of the Fifteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., 1998, pp. 144-151.
- [16] W. Buntine, T. Niblett, A further comparison of splitting

- rules for decision-tree induction, *Machine Learning* 8 (1992) 75-85.
- [17] E. Rahm, P.A. Bernstein, A survey of approaches to automatic schema matching, *The VLDB Journal* 10 (2001) 334-350.
- [18] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization, *Machine Learning* 40 (2000) 139-157.
- [19] S. Garcia, A. Fernandez, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability, *Soft Computing* 13 (2009) 959-977.
- [20] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, *Neural Computation* 10 (1998) 1895-1923.