Automation in Construction 68 (2016) 146–155

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Matching experts' decisions in concrete delivery dispatching centers by ensemble learning algorithms: Tactical level

Mojtaba Maghrebi [a,*], Travis Waller [b], Claude Sammut [c]

[a] School of Civil and Environmental Engineering, University of New South Wales (UNSW), Sydney, Australia and Department Civil Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Iran
[b] School of Civil and Environmental Engineering, University of New South Wales (UNSW), Sydney, Australia
[c] School of Computer Science Engineering, University of New South Wales (UNSW), Sydney, Australia

ARTICLE INFO

Article history:
Received 25 September 2014
Received in revised form 24 February 2016
Accepted 13 March 2016
Available online 7 April 2016

Keywords:
Ready Mixed Concrete (RMC)
Machine learning
Dispatching

ABSTRACT

Ready Mixed Concrete (RMC) suffers from a lack of practical solutions for automatic resource allocation. Under these circumstances, RMC dispatching systems are mostly handled by experts. This paper attempts to introduce a machine learning based method to automatically match experts' decisions in RMC. For this purpose, seven machine learning techniques with their boosted algorithms were selected. A set of attributes was extracted from the collected field data. Eleven metrics were used to assess the performance of the selected techniques using different approaches. Due to concerns about randomness, significant testing was performed to assist in finding the best algorithm for this purpose. Results show that Random-Forest with 85% accuracy outperforms the other selected techniques. One of the most interesting achieved results is related to the computing time. The results show that all the selected algorithms can solve large-scale depot allocations with a very short computing time. This is possibly because a model built by a machine learning algorithm only needs to be tested with new instances, which does not need an extensive computation effort. This provides us with a chance to move toward automation in Ready Mixed Concrete Dispatching Problems (RMCDPs), especially for those RMCs with dynamic environments where resource allocation might need to be quickly recalculated during the RMC process due to changes in the system.

1. Introduction

As was extensively discussed in [1], in an RMC batch plant, based on orders, the specifications of concrete mix are designed and raw materials are mixed together. Then fresh concrete is loaded into a truck. The loaded truck hauls the concrete and pours it at the destination and then returns to the batch plant. In practice, the mixing part is performed automatically; however, the rest of the process is handled by human experts. In detail, dispatchers decide to send a truck from a batch plant at a specific time to a project. This job becomes more complicated when a dispatcher needs to make calculated decisions for supplying concrete for a certain project that is located between two or more batch plants. The dispatcher needs to consider many parameters that can be categorized into three types of information: (i) specification of each order, (ii) travel of truck(s), and (iii) batch plant limitations. Moreover, a Ready Mixed Concrete Dispatching Problem (RMCDP) can be modeled as a network where customers and depots are its nodes and deliveries are the arches between depots and customers. 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.

In the last twenty years, researchers have investigated a variety of approaches to improve the efficiency of RMCDP. However, despite substantial developments in this area, RMCDP still suffers from a lack of practical solutions [2–4] and this process continues to be mostly handled by experts [3,5]. The first drawback of such a human intensive system is its dependence on the human resources, regardless of the quality of the experts' decisions. The second potential problem is the unavailability of experts in some geographical regions. The third risk is related to human error, which does not allow experts to achieve better results. In current methods of RMCDP, human error is an inevitable problem and it becomes more crucial when there is no controlling system for the experts' decisions [3]. The last and also main threat for RMCDP is the lack of automated processes. This can be critical when the demand for concrete, regardless of geographical location, is increasing throughout the world [6–12]. Current Portland cement production throughout the world will nearly double by 2050 [13]. In this paper, we introduce an automated RMCDP method at the tactical level by looking at this problem from a new angle. As has been mentioned, experts are handling RMCDP and we are attempting to match their decisions by using ensemble machine learning algorithms. Also, the size

[*] Corresponding author at: School of Civil and Environmental Engineering, Faculty of Engineering, The University of New South Wales (UNSW), Kensington, NSW 2052, Sydney, Australia. Tel.: +61 2 9385 6914, +61 450509295; fax: +61 2 9385 6139.
E-mail address: maghrebi@unsw.edu.au (M. Maghrebi).

http://dx.doi.org/10.1016/j.autcon.2016.03.007
0926-5805/© 2016 Elsevier B.V. All rights reserved.

and amount of data that is used in this study is much greater than the datasets that have been used in similar research in the literature. The richness of the data helps the authors to draw their conclusions more confidently and introduce more generalized models. In this paper, first the context of RMCDP and the related literature are discussed. Second, in the Methodology section, the selected attributes are explained and the selected machine learning techniques are presented. In the third section, the features of the dataset acquired from the field are studied. Finally, the proposed idea is tested with field data and the results presented; the outcomes are discussed by comparing the selected machine learning techniques from a different perspective.

2. Related works

A text mining based study was performed on the body of literature that was devoted to "Ready Mixed Concrete" [14]. They showed that only in a few works the RMCDP were considered while concrete technology is the main core of research in this area. Moreover, as has been briefly stated, despite significant progress on RMC dispatching in the last two decades, many scholars have indicated the inefficiency of RMC dispatching and its dependence on human expertise [2,3,5,15]. A considerable amount of RMCDP literature has been published on mathematically modeling the RMCDP and solving the models heuristically. It has been proven that the RMCDP is an NP-hard problem [16–21]. This means that with available computing facilities we cannot solve large-scale RMCDP in polynomial time. A wide range of heuristic approaches have been implemented to address this issue and Genetic Algorithm (GA) has received more attention in the literature in comparison to other evolutionary methods [2,4,18,22–25]. Apart from GA, other methods include Particle Swarm Optimization (PSO) [26,27], Ant Colony Optimization (ACO) [28], Bee Colony Optimization (BCO) [29] and Tabu Search (TS) [29]. Although the different methods have been implemented, the discrete solution structure remains fairly much the same in most of the methods which consist of two merged parts: the first part defines the sources of deliveries and the second part expresses the priorities of customers. In the mentioned studies, the two most critical challenges are the number of infeasible allocations that exist in the initial solutions and computing time; this is because RMCDP has many side constraints that must be checked at each iteration, and it is also due to the random search behavior of the evolutionary methods. These methods mostly need a post-computing process to find feasible alternatives for infeasible allocations among the initial solutions. To overcome this issue Maghrebi, Waller and Sammut [18] presented an evolutionary based method which can solve the RMCDP without needing any additional algorithm and they developed a sequential meta-heuristic method which is 10 times faster than their previous method and rather than direct travel costs can also minimize the number of fleets [30]. More recently, Liu et al. [31] introduced an integrated framework for solving both production and delivery of RMC. Chou and Ongkowijoyo [32] present a decision aid model for selecting the on-site RMC type based on a reliability assessment process. Zhang et al. [33] integrated an intelligent monitoring system with a hybrid heuristic algorithm to more effectively reschedule RMCDP when customers' demands are assumed to be dynamic. Kinable et al. [34] introduced a new formulation similar to [16] but solved using constraint programming.

Beyond the heuristic approaches, other methods should be mentioned, such as Yan, Lai and Chen [21] who introduced a numerical method for solving the RMC optimization problem by cutting the solution space and incorporating the branch and bound technique and the linear programming method. Yan, Lin and Jiang [35] used decomposition and relaxation techniques coupled with a mathematical solver. Variable Neighborhood Search (VNS) was applied in RMCDP by Payr and Schmid [36]. More recently, Maghrebi et al. [17] implemented a Column Generation (CG) method which is amenable to Dantzig–Wolfe reformulation for solving large scale models which available computing facilities cannot optimally solve in polynomial time and this approach later on was compared with a heuristic method [37]. Similarly Benders decomposition was hired to near optimally solve RMCDP within a practical time [38]. Exploring experts' decisions in RMC dispatching centers was considered by [39].

The critical issue here is why the RMCDP is still being handled by experts. The reason behind the lack of success of the mathematically based models of RMC dispatching has been discussed in [3]. They found (i) the large number of variables and (ii) dealing with uncertain and dynamic data to be the two main obstacles in the models that attempt to solve the RMCDP optimally or near optimally. In this paper, we attempt to solve the RMCDP automatically. To this end, a wide range of machine learning techniques are used to match human decisions.

For these purposes, we intend to rely on experts' decisions in the RMC dispatching room and use their decisions for training the machine learning algorithm. A valid concern about this approach is the quality of the experts' decisions. Assessing the quality of experts' decisions is a cumbersome job. This is because: firstly, the quality of experts' decisions for large instances cannot be assessed due to the unavailability of the optimum solution for large instances; and secondly, RMC owners are pleased with the performances of experts because RMC dispatching jobs are still handled by experts who are able to find feasible solutions for day-to-day RMCDP [3]. This issue was extensively studied in [40,41] for relatively large instances (around 200 deliveries per day) by using high-performance computing facilities. They modeled RMCDP with soft time window (mixed-integer programming) and solid time window (integer programming) and tested these optimization models with different sizes of field data belonging to an active RMC. The optimization results were used as a benchmark and then compared with experts' decisions. The results show that in terms of cost, experts' decisions are around 90% accurate and are also more flexible in the event of an unexpected change in the system. They also argued that the experts' approach is totally different from the optimization models. In optimization, finding a feasible solution at the least cost is desirable, but in reality it is expected that experts supply all customers with the available resources and keep all the customers satisfied.

3. Methodology

In this paper we aim to introduce a method that can match experts' decisions automatically. In other words, we are looking for an alternative way of doing what is already done by experts in RMC dispatching rooms. To implement this idea, a wide range of supervised machine learning techniques are used. The training data includes the RMC monitoring data which covers all the information provided to the experts as well as the decisions that the experts have made. In particular, the dataset shows the experts' decisions in several circumstances. Therefore, it is expected that the selected machine learning techniques will match experts' decisions in any circumstances. In the attribute selection process, two issues have been taken into account: (i) the conducted research in this area such as [2,3,15,42–45]; and (ii) a consideration of the data that is already provided in practice to the experts which was determined after carefully observing the experts' behaviors in several RMC dispatching rooms. Then, the following parameters have been selected to construct the training and test datasets.

$$y = f(x) \tag{1}$$

y experts' decisions about a selected depot for each delivery (decision variable)
f machine learning technique (classifier)
x input attributes set includes (parameters):
DOW day of delivery in the week (Monday, Tuesday, …, Sunday)
VOD volume of delivery (m^3)
EAT expected arrival time at customer (hh:mm)
LON longitude of customer

LAT	latitude of customer
TNO	total number of received orders in day
NCC	number of close customers to each customer

For example in a delivery, *y* shows the source of the delivery or particularly a depot's ID that an expert picked for this delivery and consists of a number of values that listed in Eq. (1) including the day of delivery (e.g. Monday), volume of delivery (e.g. 5.8 m^3), expected arrival time (e.g. 10:15 am), longitude (e.g. 151.231436), latitude (e.g. −33.915630), total number of received orders by the RMC (e.g. 160) and number of close customers to this customer in this day (e.g. 95).

LON and *LAT* follow the geographic coordinate system. For calculating *NCC*, in a day the number of customers for whom D_i is their nearest source is counted. This value is assigned to *NCC* for those instances when D_i is their closest depot. For example, the nearest depot to customer *i* is depot *j* when for another *n* customers depot *j* is also the nearest depot, so in this case *NCC* is equal to *n* which reflects the level of demand around each depot and also shows the density of customers around a customer in a day. The training set that includes $\{x,y\}$ is constructed from a real database whose features are studied below.

3.1. Selected machine learning techniques

Seven machine learning techniques were selected to test the proposed idea with the available dataset. The selected techniques are Decision Tree, Random-Forest, Rules, Artificial Neural Network, Support Vector Machine, K-Nearest Neighbors and Naive Bayes. In addition to the selected techniques, an ensemble version of each of these has been used and the results of all 14 techniques are reported in the section showing the results.

3.1.1. Decision Tree

Decision Tree is most likely the workhorse of machine learning techniques due to its wide use in practice [46]. Among Decision Tree induction techniques, the J48 was selected since theoretically it is very similar to C4.5, which was developed by Quinlan [47]. It uses a “divide-and-conquer” approach for building its structure, which consists of nodes, branches and leaves. A node tests a particular attribute and, based on the possible values for the attribute, some branches are added to the node. A leaf represents a class and when an instance reaches a leaf, the leaf's class will be assigned to the instance. The pruning process, which is mostly applied after building a Decision Tree, can prevent the technique from over-fitting problems and assists in interpreting the structure of the Decision Tree [47]. In building a Decision Tree, the first decision must be to select an attribute (as a node) and then divide it (branching). The concept of entropy [48] is used for attribute selection and for calculating the *information* provided by each attribute on a sample of training examples (s).

$$Entropy(s) = entropy(P_1, P_2, \ldots, P_c) \equiv \sum_{i=1}^{c} -P_i \log_2 P_i \tag{2}$$

where *c* is the number of values that the final class can take and P_i is the proportion of *S* that belongs to $i \le c$. Then *informationGain* is calculated for attribute *A*, when $values(A)$ is the set of values of *A* and $S_v \in S$ when the value of *A* is *v*.

$$Gain(S,A) = Entropy(s) = \sum_{v \in values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{3}$$

The main danger of any induction algorithm is overfitting which can happen if noise is present in the training set and the algorithm attempts to model the noisy examples. In other words, hypothesis $h \in S$ overfits the data if $h^* \in S$ and $error_D$ are errors in the entire data:

$$error_{train}(h) < error_{train}(h^*) \tag{4}$$

$$error_D(h) < error_D(h^*). \tag{5}$$

To avoid overfitting, *GainRation* can be used as a new metric.

$$SplitInformation(S,A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \tag{6}$$

$$GainRation(S,A) \equiv \frac{-Gain(S,A)}{SplitInformation(S,A)} \tag{7}$$

Moreover, reduced error pruning which was introduced by [49] tries to avoid overfitting. This technique allows the algorithm to grow the tree and then prunes the tree. It replaces a sub-tree on a leaf and assigns the most common class to that leaf. Then the pruned tree is tested by unseen instances. If the accuracy has not decreased then the prune is accepted. This process is iteratively applied to all sub-trees.

3.1.2. Random-Forest

Random-Forest can be categorized as ensemble learning. This technique grows many classification trees which are based on the random sampled instance. The class for each instance is chosen by voting among the constructed trees [50]. This is a powerful technique for dealing with noisy data, and is also a competitive algorithm for boosting adaptive bagging algorithms [51]. Random-Forests builds *K* tree-based learners $h_1(x), h_2(x), \ldots, h_k(x)$ and is trained with pair $(x;y)$ instances. *x* consists of independent attributes and *y* is the class label which can take *c*, a different class label. $p(y_i)$ is the probability of the class y_i and $p(y_i|\ v_{i,k})$ is the probability of class y_i conditioned by the attribute A_i having value v_k. Each tree is trained with *n* instances that are selected randomly with replacements. For splitting, a subset of attributes is selected randomly and the attribute with the highest Gini index is selected in that node. The constructed trees are not pruned and the final class label for an instance is obtained by voting from all individual trees. O_k and $\mathrm{Q}(x, y_i)$ are defined respectively as out-of-bag instances and out-of-bag proportion of class y_i votes, and also $\mathrm{I}(.)$ is defined as an indicator function.

$$\mathrm{Q}(x, y_i) = \frac{\sum_{k=1}^{K} \mathrm{I}(h_k(x) = y_i; (x,y) \in \mathrm{O}_k)}{\sum_{k=1}^{K} \mathrm{I}(h_k(x); (x,y) \in \mathrm{O}_k)} \tag{8}$$

The margin function, which measures the gap between the average vote for right class *y* and the average vote for all other classes, is defined as follows.

$$mr(x,y) = P(h(x) = y) - \ \max\nolimits_{j=1, j \ne y}^{c} P(h(\mathrm{x}) = y_i) \tag{9}$$

Strength s is also defined as the expected margin and is calculated by measuring the average over the training set.

$$\mathrm{s} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathrm{Q}(x_i, y) - \max\nolimits_{j=1,j \ne y}^{c} \mathrm{Q}(x_i, y_i)\right) \tag{10}$$

The variance of the margin over the square of the standard deviation of the series k trees is used for computing the average correlation as follows:

$$\overline{p} = \frac{var(mr)}{sd(h)^2} = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(\mathrm{Q}(x_i, y) - \ \max\nolimits_{j=1,j \ne y}^{c} \mathrm{Q}(x_i, y_i)\right)^2 - s^2}{\left(\frac{1}{k}\sum_{k=1}^{K}\sqrt{p_k + \widehat{p_k} + (p_k - \widehat{p_k})^2}\right)^2} \tag{11}$$

where p_k is an out-of-bag estimate of $\mathrm{P}(h_k(x) \ = y)$ and $\widehat{p_k}$ is an out-of-bag estimate $\mathrm{P}(h_k(x) = \widehat{y_i})$ and $\widehat{y_i}$ is an estimate for every training instance *x* with $\mathrm{Q}(x, y_i)$.

3.1.2. Rules

From rule-based techniques, the PART [52] has been selected. This "separate-and-conquer" approach uses partial C4.5 in each interaction. The main difference between Decision Tree and rule based techniques is that Decision Tree can pick several attributes at each node but the rules based technique only focuses on one class at a time [53]. PART does not need any global optimization unlike C4.5 and RIPPER [54], and can learn one rule at a time. It uses a Decision Tree for creating the rules and then removes the instances that were covered. After building a tree, the most converged leaf is selected for generating a rule. This process is repeated until no instances are left. Simplicity is the main advantage of PART in comparison to other similar rules schemes, while its performance is compatible to other C4.5 and RIPPER.

3.1.3. Multilayer Perceptron

The Multilayer Perceptron (MLP) is a feed-forward artificial neural network model which uses back-propagation for training its network [55]. The network consists of input nodes which are training attributes, internal nodes or a hidden layer, and output nodes which in this case are the decision classes and the perceptrons between them. A perceptron attempts to find a hyperplane to classify in high dimensional space [56]. MLP is very adaptive for learning, which means it is capable of learning how to find a relationship between inputs and outputs [57].

3.1.4. Support Vector Machine

A Support Vector Machine (SVM) is a statistical learning method that was developed by [58,59]. In a simple binary classification problem

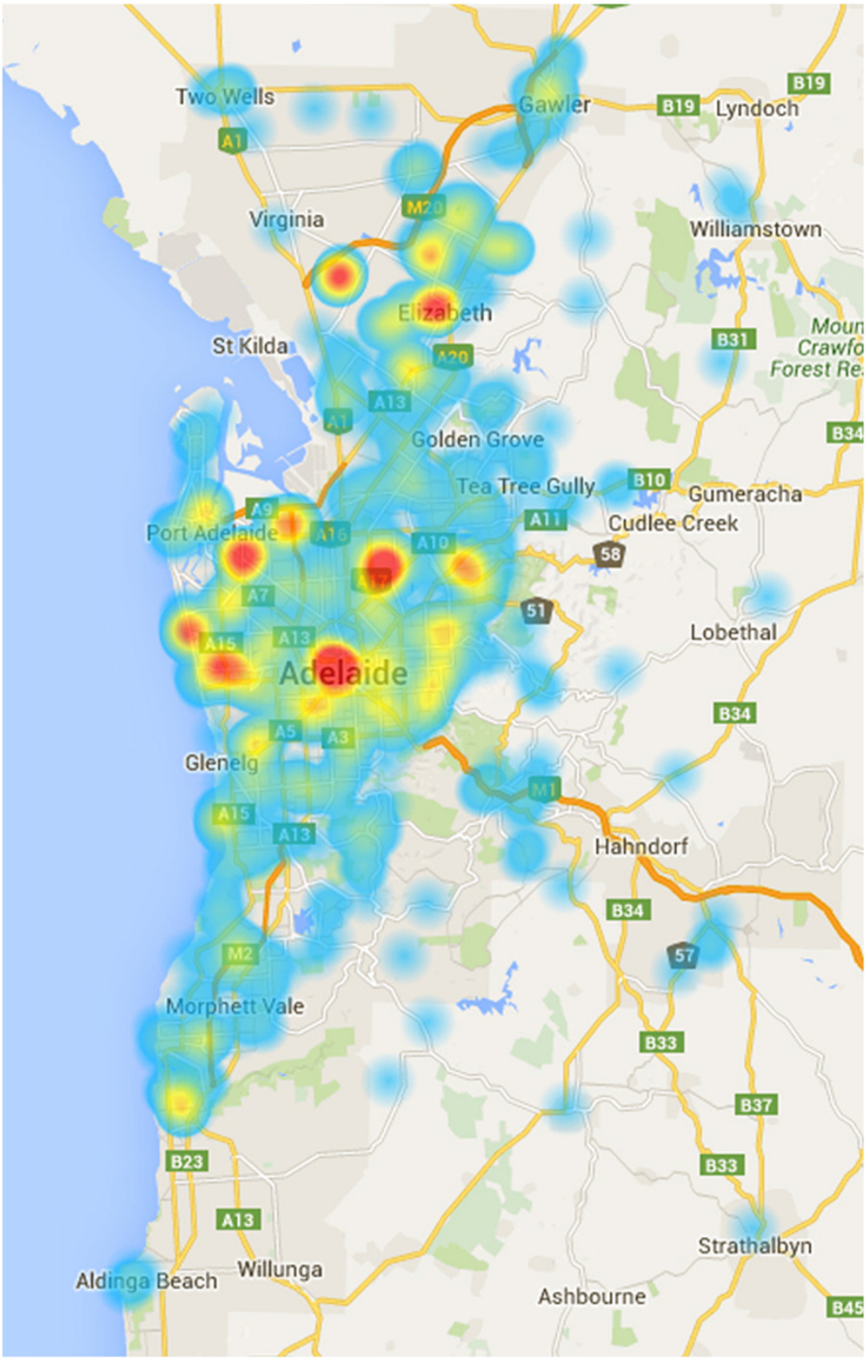

Fig. 1. Distribution of customers in the metropolitan area.

with (x,y) dataset where $y\in\{-1,+1\}$ it can separate two classes as follows using an hyperplane where w is weight vector and b is bias:

$$\langle(w,x_i)+b\rangle\geq+1 \quad \text{if} \quad y_i=+1 \tag{12}$$

$$\langle(w,x_i)+b\rangle\geq-1 \quad \text{if} \quad y_i=-1 \tag{13}$$

which can be summarized as:

$$y_i\langle(w,x_i)+b\rangle\geq+1 \tag{14}$$

the distance between a point x_i and the classifier hyperplane is

$$\frac{|\langle(w,x_i)+b\rangle\,|}{||w||} \tag{15}$$

and with margin p then

$$\frac{y_i\,|\langle(w,x_i)+b\rangle\,|}{||w||}\geq p \tag{16}$$

for the canonical hyperplane numerator is equal to one, and the distance of training instances which are closest to the hyperplane so-called *support vectors* is calculated as follows:

$$r=\frac{1}{||w||} \tag{17}$$

$$p=2r=\frac{2}{||w||}. \tag{18}$$

Thus, for finding the maximum margin the $||w||$ needs to be minimized.

$$minimize\ \frac{1}{2}\left| |w||^2\right. \tag{19}$$

Subject to:

$$y_i\langle(w,x_i)+b\rangle\geq+1 \quad i=1,\dots,n. \tag{20}$$

This model can be transformed into the dual space by using Lagrangian [60]

$$L(w,b,\alpha)=\frac{1}{2}||w||^2-\sum_{i=1}^{n}\alpha_\text{i}(y_i(\langle(w,x_i)+b\rangle)-1) \tag{21}$$

$$\frac{\partial L(w,b,\alpha)}{\partial w}=0\Rightarrow w\sum_{i=1}^{n}\alpha_i y_i x_i \tag{22}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b}=0\Rightarrow w\sum_{i=1}^{n}\alpha_i y_i=0 \tag{23}$$

and therefore can be considered as a dual optimization problem.

$$maximize\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_{j\langle x_i,x_j\rangle} \tag{24}$$

Subject to:

$$\sum_{i=1}^{n}\alpha_i y_i=0 \tag{25}$$

$$\alpha_i\geq0 \quad i=1,\dots,n \tag{26}$$

α_i so-called support vectors which follow the Karush–Kuhn–Tucker [61] theorem.

$$\alpha_i[\,y_i(\langle w,x_i\rangle+b)-1]=0 \quad i=1,\dots,n \tag{27}$$

$$\alpha_i\geq0\Longrightarrow\ y_i(\langle w,x_i\rangle+b)-1=0 \tag{28}$$

$$\alpha_i=0\Longrightarrow y_i(\langle w,x_i\rangle+b)-1\geq0 \tag{29}$$

Thus, only support vectors which have non-zero α_i and x_i's with $\alpha_i=0$ are beyond separating hyperplanes [62]. Also, the similarly model soft margin in SVM (Eq. (30)) is valid for all misclassified instances.

$$y_i(\langle w,x_i\rangle+b)\geq1-\xi_i \tag{30}$$

Thus $\xi_i\geq1$ must be minimized.

$$minimize\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i \tag{31}$$

Subject to:

$$y_i(\langle w,x_i\rangle+b)\geq1-\xi_i \quad i=1,\dots,n \tag{32}$$

$$\xi_i\geq0 \quad i=1,\dots,n \tag{33}$$

$$C\geq\alpha_i\geq0 \quad i=1,\dots,n \tag{34}$$

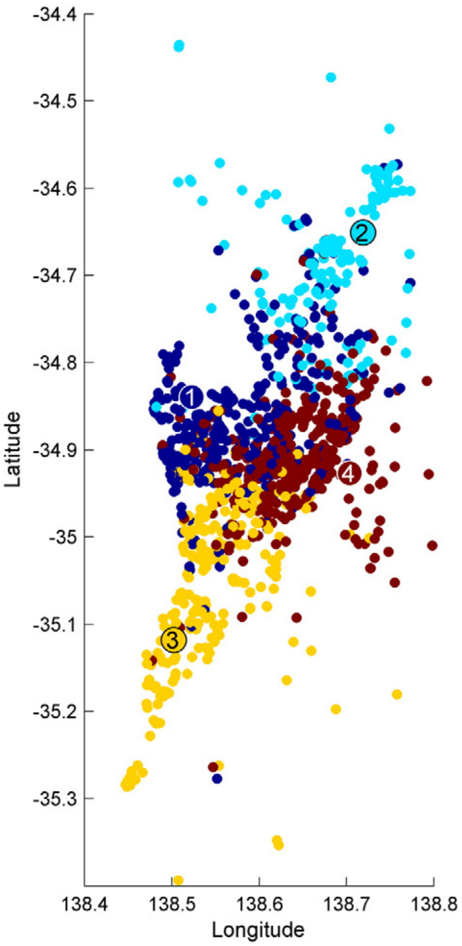

Fig. 2. Categorizing customers based on the assigned depot.

Among the SVM techniques, we picked SMO (Sequential Minimal Optimization) which was developed by Microsoft [63]. This is an SVM and is fast in solving large quadratic programming (QP) problems. It splits the QP problems into a series of simpler possible QP problems and then solves these small components analytically. SMO is able to deal with a huge number of attributes in training sets and is also flexible in avoiding overfitting by maximizing the margin around its hyperplane [64].

3.1.5. Naive Bayes

Among the probabilistic learning approaches, we selected the Naïve Bayes (NB) [65]. This is a combination of Bayes' theorem and naïve independence assumptions. It is based on an independent feature model and always prefers the simple things first. It also assumes that attributes are completely independent. Therefore, if inevitably there are dependencies between the attributes, the performance of NB will be affected.

3.1.6. Lazy (KNN)

From instance-based learning techniques, which are also called Lazy techniques, the KNN (K-Nearest Neighbors) methods are very widely used. From these techniques we selected the IBK [66] technique, which for every new testing instance only stores the training data based on neighbors and attempts to find similarities and the best class. IBK uses the distance weighting method to decrease the effects of far neighbors. Since the solution is local, the data will not be lost in generalization [67].

3.1.7. Ensemble technique

Ensemble methods are designed to multiple algorithms and obtain a better performance from any of the constituent models. Among the ensemble learning methods we selected the AdaBoost which was developed by Freund and Schapire [68]. AdaBoost works by repeatedly running a classifier over the training data and increasing the weight of misclassified instances after each run. The class of an instance is determined by voting among the built classifiers. This method forces the classifier to focus on the most difficult instances. For a simple problem $h : X \rightarrow Y$ with the training set $(x_1,y_1), \ldots ,(x_m,y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$.AdaBoost calls a learning scheme (called a weak algorithm) and repeats it T times. At run $t, t \leq T$ the weight of instance i is denoted by $D_t(i)$ which initially for all instances is set equally; however, after each run the weight of misclassified instances increases. This forces the learning scheme to focus on difficult instances.

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if} \quad h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if} \quad h_t(x_i) \neq y_i \end{cases} \\ D_{t+1}(i) &= \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \end{aligned} \tag{35}$$

when

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t} \tag{36}$$

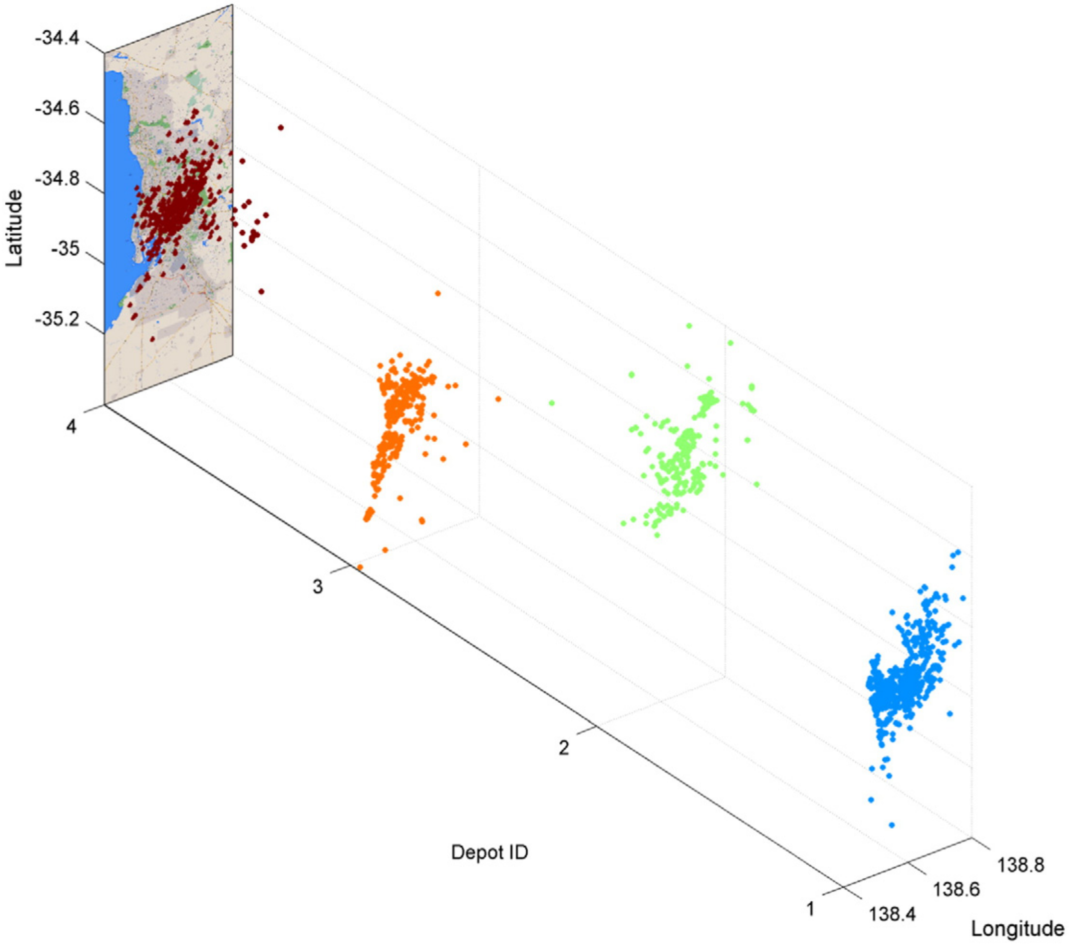

Fig. 3. Categorizing customers based on assigned depots in a 3D view.

Table 1
Summary of results.

	Algorithm ID	ACC	FSC	APR	PRC	RMSE	CVMS	MSP	AUC	SAR	Building time (s)	Testing time (s)
J48	1	83.244	0.239	0.85	0.03	0.826	0.85	0.837	0.948	0.847	0.219	0.002
Random-Forest	2	83.592	0.221	0.868	0.03	0.83	0.868	0.848	0.968	0.861	0.496	0.005
PART	3	81.577	0.247	0.833	0.03	0.823	0.833	0.828	0.942	0.837	1.83	0.012
ANN	4	77.883	0.259	0.781	0.032	0.808	0.781	0.793	0.962	0.827	20.482	0.004
SVM	5	76.664	0.328	0.722	0.024	0.839	0.722	0.775	0.921	0.786	4.249	0.005
NB	6	75.233	0.274	0.73	0.026	0.824	0.73	0.774	0.942	0.807	0.013	0.011
KNN	7	73.964	0.273	0.683	0.019	0.861	0.683	0.761	0.949	0.805	0.001	0.338
Boosted-J48	8	83.48	0.244	0.85	0.027	0.842	0.85	0.846	0.972	0.854	2.472	0.011
Boosted Random-Forest	9	84.015	0.249	0.853	0.029	0.83	0.853	0.841	0.966	0.852	4.856	0.065
Boosted PART	10	83.779	0.244	0.841	0.026	0.847	0.841	0.843	0.97	0.855	29.334	0.148
Boosted ANN	11	77.883	0.28	0.781	0.032	0.808	0.781	0.793	0.942	0.813	124.624	0.016
Boosted SVM	12	76.664	0.304	0.722	0.024	0.839	0.722	0.775	0.845	0.769	18.577	0.005
Boosted NB	13	75.233	0.311	0.73	0.026	0.824	0.73	0.774	0.88	0.774	0.772	0.037
Boosted KNN	14	73.964	0.303	0.683	0.019	0.861	0.683	0.761	0.87	0.769	22.282	1.065

ACC = accuracy.
FSC = F-score.
APR = average precision.
PRC = precision.
RMSE = root mean square error.
CVMS = cross-validation mean sensitivity.
MSP = mean specificity.
AUC = area under the ROC (receiver operating characteristic).
SAR = $\frac{ACC+AUC+(1-RMSE)}{3}$.

where Z_t is the normalization constant and ε_t is the error of h_t. The final output is calculated by voting.

$$H_{final}(x) = sign\left(\sum_t \alpha_t h_t(x)\right)$$

4. Field data

As was mentioned, the size and amount of data that is used in this study is much greater than the datasets that have been used in similar research in the literature. The richness of the data helps the authors to draw their conclusions more confidently and introduce more generalized models. The proposed approach is tested with a field dataset belonging to an active RMC which belongs to one of the largest RMC companies in Australia. For the collected dataset we particularly focused on the Adelaide metropolitan area. The available database covers 4 months in 2012 of the RMC which has 4 batch plants and around 40 trucks. In Fig. 1, the distribution of projects across the metropolitan area is illustrated, which supports the premise that customers' locations do not follow a uniform pattern.

It also might be assumed that customers in a particular area are supplied from a nearby depot, but in practice this is not always true. Figs. 2 and 3 show the complexity of depot allocation for each delivery. Fig. 2 illustrates the allocated depot for each customer by assigning different colors in order to plot customers, where each of the four colors belongs to a depot. Fig. 3 plots each color in a separate 3D plate to show the huge number of duplications in different geographical locations. This figure disproves the abovementioned assumption. The huge number of overlaps between the supply areas provides proof that probably other attributes rather than simply the location of customers are being taken into account by experts when assigning a depot to a customer.

5. Results and discussion

5.1. Evaluation

The selected learning schemes are tested with the available dataset. 10 fold cross-validation is used for evaluating the selected algorithms,

Table 2
Results of paired t-test (P values).

		Random –Forest	PART	ANN	SVM	NB	KNN	Boosted J48	Boosted Random– Forest	Boosted PART	Boosted ANN	Boosted SVM	Boosted NB	Boosted KNN
10 fold	J48	0.54676	0.06843	2.8E−06	4.6623E−08	3.7113E−09	1.0667E−10	0.64152805	0.43063309	0.4090856	0.00012295	4.6623E−08	3.7113E−09	1.0667E−10
	Random–Forest		0.01901	6.1E−07	1.0156E−08	7.7589E−10	2.2751E−11	0.40818081	0.82709411	0.8194608	2.8831E−05	1.0156E−08	7.7589E−10	2.2751E−11
	PART			7.0E−05	6.9759E−07	2.7162E−08	5.0552E−10	0.54745121	0.01483109	0.01147406	0.00187305	6.9759E−07	2.7162E−08	5.0552E−10
	ANN				0.08676506	0.00142665	1.326E−05	0.00205116	6.9216E−07	4.037E−07	0.67841977	0.08676506	0.00142665	1.326E−05
	SVM					0.0461836	0.00036912	0.00015722	1.3647E−08	6.9329E−09	0.07708727	1	0.0461836	0.00036912
	NB						0.06054278	1.3243E−05	1.0806E−09	5.5837E−10	0.00287098	0.0461836	1	0.06054278
	KNN							1.116E−06	3.6654E−11	1.6858E−11	7.58E−05	0.00036912	0.06054278	1
	Boosted–J48								0.34698716	0.34015198	0.00780035	0.00015722	1.3243E−05	1.116E−06
	Boosted– Random–Forest									0.99973213	2.6592E−05	1.3647E−08	1.0806E−09	3.6654E−11
	Boosted–PART										1.9775E−05	6.9329E−09	5.5837E−10	1.6858E−11
	Boosted–ANN											0.07708727	0.00287098	7.58E−05
	Boosted–SVM												0.0461836	0.00036912
	Boosted–NB													0.06054278

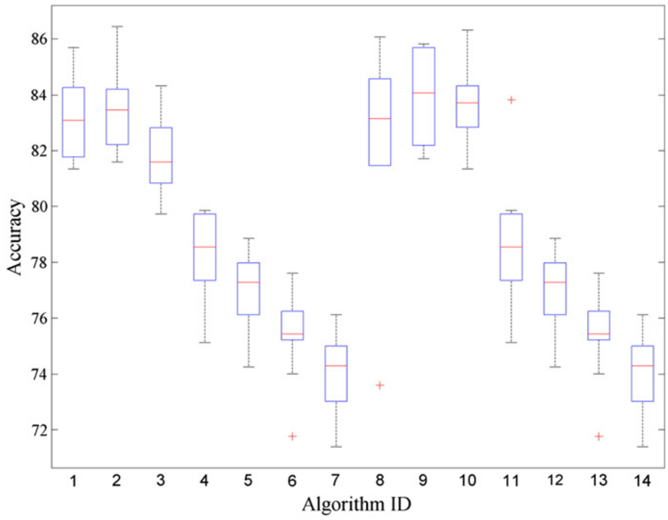

Fig. 4. Distribution of accuracies obtained from 10 fold cross-validation.

with 10 fold cross-validation being the standard way of assessing a learning scheme on a particular dataset [46]. In this evaluation method, the datasets are divided into 10 fold, with around 9 fold used for training and the remaining 10% of the data being used for testing.

5.2. Assessing metrics

Eleven metrics have been selected for assessing the performance of the selected algorithms from different perspectives which are listed in Table 1. From a general perspective, accuracy (ACC) is the most important feature when comparing two or more algorithms because it reflects the ability of each learning scheme to identify the correct class, which also is the main job of a classifier. In terms of accuracy, Boosted-Random-Forest, Boosted PART, Random-Forest, Boosted J48, J48 and PART achieved the ranks 1 to 6 respectively, with KNN being the last. The achieved results for ANN, SVM and NB and their boosted algorithms are exactly same, while in other metrics there are slight differences, particularly in time.

5.3. Algorithms ranking

As is evident in (Table 1), the accuracy rates of most of the algorithms are similar. In instances where algorithms are slightly different due to randomness concerns, those algorithms which outperform others based only on accuracy cannot be identified [69–71]. Moreover, the best algorithm in one feature may not necessarily be 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, and is used for comparing machine learning algorithms in one domain [43,72]. The t-test investigates the meaningful difference between a pair of algorithms. There are two hypotheses in a t-test: H_0 for equality and H_1 for inequality at α level of significance. The P value is calculated through a t-test. If it is lower than α then H_1 will be accepted. Otherwise, H_0 is true. This indicates that there is no significant difference between the two algorithms. The calculated error values for each algorithm were used in a paired t-test with $\alpha = 0.01$ or at the significant level of 99%; the derived results are shown in (Table 2). In this table, the significant differences are highlighted and it is obvious that most of the learning schemes are significantly different. The average of 10 times calculating accuracy obtained from 10 fold cross-validation for most of the selected algorithms are very similar. However, because their standard deviations are very small, most of the algorithms are significantly different. We have attempted to illustrate this issue in Fig. 4. According to the results in (Tables 1 and 2), we can drop Boosted J48, J48 and PART from the shortlist of the top 6 algorithms because there are significant differences between those and Boosted Random-Forest (rank 1) and Random-Forest (rank 2). However, there is no significant difference between ranks 1 to 3. Under these circumstances, and considering the elapsed time, out of the top 3 algorithms Random-Forest qualifies for further studies.

One of the most interesting of the achieved results is related to the computing time. This result shows that all of the selected algorithms can solve depot allocation for around 800 customers in less than a second. In the case of using updatable learning algorithms in which the relearning process is being done with a new instance, the computing time also is very small. This ability of machine learning provides us with an opportunity to move toward automation in RMCDP especially when the system is changed and re-allocation must be done. Moreover, from these results it can clearly be seen that the machine learning approach can match experts' decisions with an accuracy of around 85%. The issue that must be considered here is human error. It is expected from the qualified machine learning algorithms that any human error will be detected and a better solution can be provided. So, perhaps the misclassified instances which are counted as errors are not truly bad decisions. Another possibility is that under some circumstances the experts are free to choose a depot from a list of available depots, and their decisions are mostly arbitrary. This means that another choice from the list of available depots can be an acceptable decision. This topic needs an extensive coding process to investigate the feasibility of misclassified instances by the qualified machine learning scheme (Random-Forest) that can be considered as future work.

6. Conclusion

Experts generally handle Ready Mixed Concrete Dispatching Problems (RMCDPs) manually by taking into account the real data that is mostly supplied for them automatically. This paper has introduced a new approach for the automatic allocation of the resource at the tactical level of RMC dispatching. To do so, the depot assignment task was formulated as a supervised machine learning problem (classification) which can be trained by observing experts' decisions in RMC dispatching rooms. For this purpose, seven machine learning techniques with their boosted algorithms were selected and compared with each other. A set of attributes was extracted from a field dataset to construct the dataset. This database was gathered for a period of 4 months from an active batch plant in Adelaide, Australia which has 4 depots and around 40 trucks. Eleven metrics were used to deeply investigating the performance of the selected machine learning techniques from different practical prospective. The significant test was conducted because most of the algorithms achieved similar results. Random-Forest with 85% accuracy outperformed other techniques by considering the t-test results and elapsed time. One of the most interesting achieved results is related to the computing time. The results show that all the selected algorithms can solve large-scale depot allocation with a very short computing time. This is possible because a model built by a machine learning algorithm only needs to be tested with new instances, which does not need an extensive computation effort. This provides us with a chance to move toward automation in RMCDP, especially for RMC with dynamic environments (e.g. order cancellation, road traffic, truck breakdown, depot breakdown) where resource allocation might need to be quickly recalculated during the RMC process due to changes in the system. Under these circumstances and when on-time decisions are required, the introduced machine learning based method can be considered as a practical decision support tool due to its short computing time as well as the ability of matching experts' decisions with a high accuracy.

In this paper, automatically solving RMCDP at a tactical level was studied using machine learning algorithms trained by experts' decisions. For future work, potential enhancement of the introduced method with operation research methodology is a consideration. For instance, there is a valid concern regarding about the feasibility of misclassified solutions (i.e., the machine learning approach might have rejected a particular solution even though it could have been made to work). Solution feasibility can be difficult to address within a pure machine learning context but a combined methodology (mixing machine learning with optimization tools) could yield advantages in terms of reducing the quantity and impact of misclassified solutions.

Acknowledgment

We acknowledge computing time at the Leonardi cluster in the Faculty of Engineering, the University of New South Wales (UNSW) and NICTA is funded by the Australian Department of Communications and the Australian Research Council through the ICT Centre of Excellence program.

References

[1] M. Maghrebi, S.T. Waller, C. Sammut, Feasibility study of automatically performing the concrete delivery dispatching by implementing machine learning techniques, Eng. Constr. Archit. Manag. 22 (5) (2015) 573–590, http://dx.doi.org/10.1108/ECAM-06-2014-0081.

[2] C.-W. Feng, T.-M. Cheng, H.-T. Wu, Optimizing the schedule of dispatching RMC trucks through genetic algorithms, Autom. Constr. 13 (3) (2004) 327–340, http://dx.doi.org/10.1016/j.autcon.2003.10.001.

[3] M. Maghrebi, C. Sammut, S.T. Waller, Reconstruction of an expert's decision making expertise in concrete dispatching by machine learning, J. Civ. Eng. Archit. 7 (12) (2013) 1540–1547, http://dx.doi.org/10.17265/1934-7359/2013.12.009.

[4] D. Naso, M. Surico, B. Turchiano, U. Kaymak, Genetic algorithms for supply-chain scheduling: a case study in the distribution of ready-mixed concrete, Eur. J. Oper. Res. 177 (3) (2007) 2069–2099, http://dx.doi.org/10.1016/j.ejor.2005.12.019.

[5] P.C. Lin, J. Wang, S.H. Huang, Y.T. Wang, Dispatching ready mixed concrete trucks under demand postponement and weight limit regulation, Autom. Constr. 19 (6) (2010) 798–807, http://dx.doi.org/10.1016/j.autcon.2010.05.002.

[6] T. Armstrong, The Global Cement Report, 10 ed. International Cement Review, London, UK, 2013 (http://www.ficem.org/boletines/ct-2013/presentaciones2013/1-EXPERTOS/2_THOMAS-ARMSTRONG/ICR-FICEM-Presentation-Handout-30Aug13.pdf).

[7] J.S. Damtoft, J. Lukasik, D. Herfort, D. Sorrentino, E.M. Gartner, Sustainable development and climate change initiatives, Cem. Concr. Res. 38 (2) (2008) 115–127, http://dx.doi.org/10.1016/j.cemconres.2007.09.008.

[8] K. Humphreys, M. Mahasenan, Towards a Sustainable Cement Industry, Climate Change, Sub-study 8, World Business Council for Sustainable Development, 2002 (http://www.cement.ca/images/stories/wbcsd-batelle_2002_climate_change_-_substudy_8.pdf).

[9] M.S. Imbabi, C. Carrigan, S. McKenna, Trends and developments in green cement and concrete technology, Int. J. Sustain. Built Environ. 1 (2) (2012) 194–216, http://dx.doi.org/10.1016/j.ijsbe.2013.05.001.

[10] N. Mahasenan, S. Smith, K. Humphreys, Y. Kaya, The cement industry and global climate change: current and potential future cement industry CO_2 emissions, in: J. Gale, Y. Kaya (Eds.), 6th International Conference on Greenhouse Gas Control Technologies, Pergamon, Oxford, UK, ISBN: 0-08-044276-5 2003, pp. 995–1000.

[11] E. Rosenthal, Cement Industry Is at Center of Climate Change Debate, New York Times, 26 October 2007 (http://www.nytimes.com/2007/10/26/business/worldbusiness/26cement.html?_r=0).

[12] E. Worrell, L. Price, N. Martin, C. Hendriks, L.O. Meida, Carbon dioxide emissions from the global cement industry, Annu. Rev. Energy Environ. 26 (1) (2001) 303–329, http://dx.doi.org/10.1146/annurev.energy.26.1.303.

[13] H. Klee, Cement industry energy and CO_2 performance: getting the numbers right, World Business Council for Sustainable Development, Cement Sustainability Initiative (CSI), ISBN: 978-3-940388-48-3, 2009 (http://www.wbcsdcement.org/pdf/CSI%20GNR%20Report%20final_updated%20Nov11_LR.pdf).

[14] M. Maghrebi, S.T. Waller, C. Sammut, Text mining approach for reviewing the ready mixed concrete literature, in: N. Yabuki (Ed.), 2nd International Conference on Civil and Building Engineering Informatics, University of Osaka, Tokyo, Japan, ISBN: 978-4-9907371-1-5 2015, pp. 105–109 http://www.see.eng.osaka-u.ac.jp/seeit/iccbei2015/pages/abstract/125.html).

[15] N.F. Matsatsinis, Towards a decision support system for the ready concrete distribution system: a case of a Greek company, Eur. J. Oper. Res. 152 (2) (2004) 487–499, http://dx.doi.org/10.1016/S0377-2217(03)00039-0.

[16] L. Asbach, U. Dorndorf, E. Pesch, Analysis, modeling and solution of the concrete delivery problem, Eur. J. Oper. Res. 193 (3) (2009) 820–835, http://dx.doi.org/10.1016/j.ejor.2007.11.011.

[17] M. Maghrebi, V. Periaraj, S.T. Waller, C. Sammut, Using column generation for solving large scale concrete dispatching problems, Comput. Aided Civ. Infrastruct. Eng. 31 (2) (2015) 145–159, http://dx.doi.org/10.1111/mice.12182.

[18] M. Maghrebi, S.T. Waller, C. Sammut, Scheduling concrete delivery problems by a robust meta heuristic method, in: D. Al-Dabass (Ed.), 7th European Symposium on Computer Modeling and Simulation (EMS), ACM, Manchester, UK 2013, pp. 354–359, http://dx.doi.org/10.1109/EMS.2013.64.

[19] V. Schmid, K.F. Doerner, R.F. Hartl, J.J. Salazar-González, Hybridization of very large neighborhood search for ready-mixed concrete delivery problems, Comput. Oper. Res. 37 (3) (2010) 559–574, http://dx.doi.org/10.1016/j.cor.2008.07.010.

[20] S. Yan, W. Lai, An optimal scheduling model for ready mixed concrete supply with overtime considerations, Autom. Constr. 16 (6) (2007) 734–744, http://dx.doi.org/10.1016/j.autcon.2006.12.009.

[21] S. Yan, W. Lai, M. Chen, Production scheduling and truck dispatching of ready mixed concrete, Transp. Res. E Logist. Transp. Rev. 44 (1) (2008) 164–179, http://dx.doi.org/10.1016/j.tre.2006.05.001.

[22] M. Cao, M. Lu, J.P. Zhang, Z.-Q. Liu, X.-Z. Wang, H. Yan, Concrete plant operations optimization using combined simulation and genetic algorithms, in: D. Yeung (Ed.), International Conference on Machine Learning and Cybernetics, Vol. No. 7, IEEE, Guanzhou, China, ISBN: 0-7803-8403-2 2004, pp. 4204–4209, http://dx.doi.org/10.1109/ICMLC.2004.1384577.

[23] J.M. Garcia, S. Lozano, K. Smith, T. Kwok, G. Villa, in: L. Wang, J. Rajapakse, K. Fukushima, S. Lee, X. Yao (Eds.), Coordinated Scheduling of Production and Delivery from Multiple Plants and with Time Windows Using Genetic Algorithms, 9th International Conference on Neural Information Processing, Vol. No. 3, IEEE, Singapore 2002, pp. 1153–1158, http://dx.doi.org/10.1109/ICONIP.2002.1202802.

[24] M. Lu, H.C. Lam, Optimized concrete delivery scheduling using combined simulation and genetic algorithms, in: M. Kuhl, N. Steiger, B. Armstrong, J. Joines (Eds.), 37th Winter Simulation Conference, ACM, Orlando, FL, USA, ISBN: 0-7803-9519-0, 2005, pp. 2572–2580.

[25] M. Lu, H.C. Lam, Simulation–optimization integrated approach to planning ready mixed concrete production and delivery: validation and applications, in: M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin, R.G. Ingalls (Eds.), 2009 Winter Simulation Conference, ACM, Austin, Texas, USA 2009, pp. 2593–2604, http://dx.doi.org/10.1109/WSC.2009.5429729.

[26] M. Lu, D.P. Wu, J.P. Zhang, A particle swarm optimization-based approach to tackling simulation optimization of stochastic, large-scale and complex systems, in: D. Liu, Z. Wang, H. Yan (Eds.), Advances in Machine Learning and Cybernetics, Springer, Guangzhou, China, ISBN: 978-3-540-33584-9 2005, pp. 528–537, http://dx.doi.org/10.1007/11739685_55.

[27] D.P. Wu, M. Lu, J.P. Zhang, in: J.W.T. Lee (Ed.), Efficient Optimization Procedures for Stochastic Simulation Systems, International Conference on Machine Learning and Cybernetics, Vol. No. 5, IEEE, Guangzhou, China 2005, pp. 2895–2900, http://dx.doi.org/10.1109/ICMLC.2005.1527437.

[28] C.A. Silva, J.M. Faria, P. Abrantes, J.M.C. Sousa, M. Surico, D. Naso, Concrete delivery using a combination of GA and ACO, in: R. Tempo, E.F. Camacho (Eds.), 44th IEEE European Conference on Decision and Control, ACM, Seville, Spain 2005, pp. 7633–7638, http://dx.doi.org/10.1109/CDC.2005.1583394.

[29] S. Srichandum, T. Rujirayanyong, Production scheduling for dispatching ready mixed concrete trucks using bee colony optimization, Am. J. Eng. Appl. Sci. 3 (1) (2010) 823–830, http://dx.doi.org/10.3844/ajeassp.2010.7.14 (ISSN: 1941-7020).

[30] M. Maghrebi, S.T. Waller, C. Sammut, Sequential meta-heuristic approach for solving large-scale ready-mixed concrete dispatching problems, Comput. Civ. Eng. 30 (1) (2014) 040141171–0401411711, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000453.

[31] Z. Liu, Y. Zhang, M. Li, Integrated scheduling of ready-mixed concrete production and delivery, Autom. Constr. 48 (2014) 31–43, http://dx.doi.org/10.1016/j.autcon.2014.08.004.

[32] J.-S. Chou, C.S. Ongkowijoyo, Reliability-based decision making for selection of ready-mix concrete supply using stochastic superiority and inferiority ranking method, Reliab. Eng. Syst. Saf. 137 (2015) 29–39, http://dx.doi.org/10.1016/j.ress.2014.12.004.

[33] G. Zhang, J. Zeng, Ready mixed concrete vehicle rescheduling method based on the internet of things, Sens. Lett. 12 (2) (2014) 431–438, http://dx.doi.org/10.1166/sl.2014.3246.

[34] J. Kinable, T. Wauters, G.V. Berghe, The concrete delivery problem, Comput. Oper. Res. 48 (0) (2014) 53–68, http://dx.doi.org/10.1016/j.cor.2014.02.008.

[35] S. Yan, H.C. Lin, X.Y. Jiang, A planning model with a solution algorithm for ready mixed concrete production and truck dispatching under stochastic travel times, Eng. Optim. 44 (4) (2012) 427–447, http://dx.doi.org/10.1080/0305215X.2011.580741.

[36] F. Payr, V. Schmid, Optimizing deliveries of ready-mixed concrete, in: M. Affenzeller, W. Jack (Eds.), 2nd International Conference on Logistics and Industrial Informatics, IEEE, Linz, Austria 2009, pp. 1–6, http://dx.doi.org/10.1109/LINDI.2009.5258569.

[37] M. Maghrebi, V. Periaraj, S.T. Waller, C. Sammut, Solving ready mixed concrete delivery problems: evolutionary comparison between column generation and robust genetic algorithm, in: R. Issa (Ed.), International Conference on Computing in Civil and Building Engineering, ASCE, Orlando, Florida, USA 2014, pp. 1417–1424, http://dx.doi.org/10.1061/9780784413616.176.

[38] M. Maghrebi, V. Periaraj, S.T. Waller, C. Sammut, Using Benders decomposition for solving ready mixed concrete dispatching problems, in: A. Akbarnejad (Ed.), 31th International Symposium on Automation and Robotics in Construction and Mining, IAARC, Sydney, Australia 2014, pp. 145–154 http://www.iaarc.org/publications/fulltext/isarc2014_submission_171.pdf).

[39] M. Maghrebi, S.T. Waller, Exploring experts decisions in concrete delivery dispatching systems using Bayesian network learning techniques, in: D. Al-Dabass (Ed.), 2nd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), IEEE, Madrid, Spain 2014, pp. 103–108, http://dx.doi.org/10.1109/AIMS.2014.9.

[40] M. Maghrebi, S.T. Waller, C. Sammut, Assessing the accuracy of expert-based decisions in dispatching ready mixed concrete, J. Constr. Eng. Manag. 140 (6) (2014) http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000853.

[41] M. Maghrebi, S.T. Waller, C. Sammut, Optimality gap of experts' decisions in concrete delivery dispatching, J. Build. Eng. 2 (1) (2015) 17–23, http://dx.doi.org/10.1016/j.jobe.2015.04.001.

[42] C.W. Feng, H.T. Wu, Integrating fmGA and CYCLONE to optimize the schedule of dispatching RMC trucks, Autom. Constr. 15 (2) (2006) 186–199, http://dx.doi.org/10.1016/j.autcon.2005.04.001.

[43] Y. Zhang, M. Li, Z. Liu, Vehicle scheduling and dispatching of ready mixed concrete, in: N. Zhang (Ed.), Fourth International Workshop on Advanced Computational Intelligence (IWACI), IEEE, Wuhan, China 2011, pp. 465–472, http://dx.doi.org/10.1109/IWACI.2011.6160052.

[44] L.D. Graham, D.R. Forbes, S.D. Smith, Modeling the ready mixed concrete delivery system with neural networks, Autom. Constr. 15 (5) (2006) 656–663, http://dx.doi.org/10.1016/j.autcon.2005.08.003.

[45] I.D. Tommelein, A. Li, Just-in-time concrete delivery mapping alternatives for vertical supply chain integration, in: I.D. Tommelein (Ed.), 7th Annual Conference of the International Group for Lean Construction IGLC-7, ACM, California, USA, 1999 (http://www.ce.berkeley.edu/~tommelein/IGLC-7/PDF/TommeleinandLi.pdf.

[46] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, New York, USA, 2011 (ISBN-10: 0123748569).

[47] R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, California, USA, 1992 (ISSN: 1049-1910).

[48] C.E. Shannon, W. Weaver, The Mathematical Theory of Communication, University of Illinois Press, Urbana, Illinois, USA, 1949 (http://www.press.uillinois.edu/books/catalog/67qhn3ym9780252725463.html).

[49] J.R. Quinlan, Simplifying decision trees, Int. J. Man Mach. Stud. 27 (3) (1987) 221–234, http://dx.doi.org/10.1016/S0020-7373(87)80053-6.

[50] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324.

[51] M. Robnik-Šikonja, Improving random forests, in: J. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), 15th European Conference on Machine Learning, Springer, Pisa, Italy 2004, pp. 359–370, http://dx.doi.org/10.1007/978-3-540-30115-8_34.

[52] E. Frank, I.H. Witten, Generating accurate rule sets without global optimization, in: J.W. Shavlik (Ed.), 15th International Conference on Machine Learning (ICML), ACM, Madison, Wisconsi, USA 1998, pp. 144–151 http://hdl.handle.net/10289/1047).

[53] W. Buntine, T. Niblett, A further comparison of splitting rules for decision-tree induction, Mach. Learn. 8 (1) (1992) 75–85, http://dx.doi.org/10.1023/A:1022686419106.

[54] W.W. Cohen, Y. Singer, A simple, fast and effective rule learner, in: J. Hendler, D. Subramanian, R. Uthurusamy, B. Hayes-Roth (Eds.), National Conference on Artificial Intelligence, AAAI, Orlando, USA 1999, pp. 335–342 (http://aaaipress.org/Papers/AAAI/1999/AAAI99-049.pdf).

[55] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Internal Representations by Error Propagation, Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, MIT Press 1986, pp. 318–362 (http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2anddoc=GetTRDoc.pdfand AD=ADA164453).

[56] S.I. Gallant, Perceptron-based learning algorithms, IEEE Trans. Neural Netw. 1 (2) (1990) 179–191, http://dx.doi.org/10.1109/72.80230.

[57] J.W. Park, R.G. Harley, G.K. Venayagamoorthy, Adaptive-critic-based optimal neurocontrol for synchronous generators in a power system using MLP/RBF neural networks, IEEE Trans. Ind. Appl. 39 (5) (2003) 1529–1540, http://dx.doi.org/10.1109/TIA.2003.816493.

[58] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: D. Haussler (Ed.), Fifth Annual Workshop on Computational Learning Theory, ACM, Pittsburgh, USA 1992, pp. 144–152, http://dx.doi.org/10.1145/130385.130401.

[59] V.N. Vapnik, S. Kotz, Estimation of Dependences Based on Empirical Data, Springer-Verlag, New York, 1982, ISBN 978-0-387-30865-4 http://dx.doi.org/10.1007/0-387-34239-7.

[60] D.G. Luenberger, Y. Yinyu, Linear and Nonlinear Programming, Springer, New York, USA, 2008, ISBN 978-0-387-74502-2 http://dx.doi.org/10.1007/978-0-387-74503-9.

[61] W. Karush, Minima of Functions of Several Variables with Inequalities as Side Constraints(Master's thesis) Dept. of Mathematics, Univ. of Chicago, 1939.

[62] M. Gönen, E. Alpaydın, Multiple kernel learning algorithms, J. Mach. Learn. Res. 12 (1) (2011) 2211–2268 (ISSN: 1532-4435, http://users.ics.aalto.fi/gonen/files/gonen_jmlr11_paper.pdf).

[63] J.C. Platt, Fast training of support vector machines using sequential minimal optimization, Advances in Kernel Methods, MIT Press, ISBN: 0-262-19416-3 1999, pp. 185–208 (http://research.microsoft.com:8082/apps/pubs/default.aspx?id=68391).

[64] I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun, Support vector machine learning for interdependent and structured output spaces, in: E.C. Bradley (Ed.), 21st International Conference on Machine Learning, ACM, Banff, Alberta, Canada, ISBN: 1-58113-838-5 2004, pp. 1–8.

[65] G.H. John, P. Langley, Estimating continuous distributions in Bayesian classifiers, in: P. Besnard, S. Hanks (Eds.),11th Conference on Uncertainty in Artificial Intelligence 1995, pp. 338–345 (http://web.cs.iastate.edu/~honavar/bayes-continuous.pdf).

[66] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1) (1991) 37–66, http://dx.doi.org/10.1007/BF00153759.

[67] R.L. De Mantaras, E. Armengol, Machine learning from examples: inductive and lazy methods, Data Knowl. Eng. 25 (1–2) (1998) 99–123, http://dx.doi.org/10.1016/S0169-023X(97)00053-0.

[68] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: L. Saitta (Ed.), International Conference on Machine Learning, UCSD, Bari, Italy 1996, pp. 148–156 (http://www.public.asu.edu/~jye02/CLASSES/Fall-2005/PAPERS/boosting-icml.pdf).

[69] E. Rahm, P.A. Bernstein, A survey of approaches to automatic schema matching, Int. J. Very Large Data Bases 10 (4) (2001) 334–350, http://dx.doi.org/10.1007/s007780100057.

[70] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization, Mach. Learn. 40 (2) (2000) 139–157, http://dx.doi.org/10.1023/A:1007607513941.

[71] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft. Comput. 13 (10) (2009) 959–977, http://dx.doi.org/10.1007/s00500-008-0392-y.

[72] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput. 10 (7) (1998) 1895–1923, http://dx.doi.org/10.1162/089976698300017197.