COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Computer-Aided Civil and Infrastructure Engineering **00** (2017) 1–18

Prediction of Bus Travel Time Using Random Forests Based on Near Neighbors

Bin Yu

School of Transportation Science and Engineering, Beihang University, Beijing, PR China and Transportation Management College, Dalian Maritime University, Dalian, PR China

Huaizhu Wang

Transportation Management College, Dalian Maritime University, Dalian, PR China

Wenxuan Shan

School of Transportation Science and Engineering, Beihang University, Beijing, PR China

&

Baozhen Yao*

School of Automotive Engineering, Dalian University of Technology, Dalian, PR China

Abstract: *The prediction of bus arrival time is important for passengers who want to determine their departure time and reduce anxiety at bus stops that lack timetables. The random forests based on the near neighbor (RFNN) method is proposed in this article to predict bus travel time, which has been calibrated and validated with real-world data. A case study with two bus routes is conducted, and the proposed RFNN is compared with four methods: linear regression (LR), k-nearest neighbors (KNN), support vector machine (SVM), and classic random forest (RF). The results indicate that the proposed model achieves high accuracy. That is, one bus route has the results of 13.65 mean absolute error (MAE), 6.90% mean absolute percentage error (MAPE), 26.37 root mean squared error (RMSE) and 13.77 (MAE), 7.58% (MAPE), 29.01 (RMSE), respectively. RFNN has a longer computation time of 44,301 seconds for a data set with 14,182 data. The proposed method can be optimized by the technology of parallel computing and can be applied to real-time prediction.*

1 INTRODUCTION

In Intelligent Transportation Systems (ITSs) and Advanced Traveler Information Systems (ATISs), the prediction of bus travel time with reasonable accuracy is important. Travelers can efficiently arrange their schedules and reduce their waiting time, if they can obtain accurate bus arrival information, because waiting time is more significant to a person than travel time (Ben-Akiva and Lerman, 1985). The prediction of bus arrival/running time is also important for bus operators. The prediction results can provide information about the future conditions of a bus system in the short term; bus operators can adjust their bus schedules by applying a higher or lower speed in advance. The prediction of bus arrival time can reduce the waste of bus resources. For example, bus travelers can change their travel mode from bus to taxi or private car when they feel anxious about bus delays without a reasonable expected arrival time, especially for bus stops without timetables.

*To whom correspondence should be addressed. E-mail: *yaobaozhen@hotmail.com*.

DOI: 10.1111/mice.12315

The objective of the prediction of bus travel time is to forecast the travel times of buses between two locations (e.g., two bus stops). Traditional time-series models aim to capture the characteristics of bus travel time over time, which only requires travel time. In addition, Automatic Passenger Counter (APC) and Automatic Vehicle Location (AVL) data are usually employed to predict bus travel time (Shalaby and Farhan, 2003). Studies also seek relations between bus travel time and passenger flows, which are collected from APC data. In these APC-based methods, the travel times of buses are assumed to be influenced by passenger flows at bus stops. Meanwhile, AVL data is useful for prediction because an AVL system provides the current locations of buses, which can be applied to the prediction of the remaining travel. In our study, we collect the data of buses from the AVL system and use this data to measure current traffic conditions on route segments.

A variety of prediction models for bus arrival/running time have been developed over the past decades. Historical average models (Jeong, 2005; Williams and Hoel, 2003), nonparametric regression models (Chan et al., 2009; Chang et al., 2010; Park et al., 2007; Smith et al., 2002; Tam and Lam, 2009), time series models (Al-Deek et al., 1998; Thomas et al., 2010; Chien et al., 2002; Kalman, 1960; Shalaby and Farhan, 2003; Yu et al., 2010), artificial neural network (ANN) models (Adeli, 2001; Adeli and Hung, 1994), and support vector machine (SVM) models (Yu et al., 2010; Yu et al., 2006) are commonly used.

Historical average models assume that historical data are similar to real-time data and predict the running time of a particular trip by averaging over several previous trips. These models may be unreliable when real-time data differ from historical data in spatial or temporal aspects.

Nonparametric regression models are simple because of the absence of estimating parameters. In the last decade, the *k*-nearest neighbor (KNN) model was extensively employed in many fields as a nonparametric regression model. Chang et al. (2010) developed a KNN model to estimate bus travel time; the results proved that the model is effective according to the accuracy and computing time of prediction.

Time series models assume a trend with time in the data set and speculate the predicted value from the trend; thus, these models are very dependent on the similarity between historical prediction and real-time prediction. The methods usually have a short time lag when applied in real-time prediction. Kalman filtering models have the ability to accommodate traffic fluctuations with time-dependent parameters. Originating from the state-space representations in modern control theory, Kalman filter model is applied for predicting short-term traffic demand and travel times. Chien and Kuchipudi (2003) developed a path-based and a link-based model to predict bus arrival time. Shalaby and Farhan (2003) proposed a Kalman filter model to predict bus arrival time and discovered that Kalman filter model outperformed the regression and ANN (artificial neural network) models. An enhanced algorithm based on Kalman filter model was developed to predict bus arrival time and was proved more effective than the standard ANN models (Chen et al., 2004).

Artificial neural network models are widely used in transportation (Adeli and Yeh, 1990; Dharia and Adeli, 2003; Jiang and Adeli, 2004, 2005; Park et al., 1991; Wu and Adeli, 2001) for its ability to deal with complex relationships in data sets. Unlike multivariable models, ANNs can be developed without specifying the form of the function, whereas the restrictions on the multicollinearity of the explanatory variables can be neglected. Chien et al. (2002) provided two ANN models to predict transit arrival time, which are trained by link-based and stop-based data. Both of the models were integrated with an adaptive algorithm to improve prediction accuracy. Meanwhile, hybrid models with a combination of Kalman filtering and neural networks showed good results (Chen et al., 2004; Chien et al., 2002). Jeong and Rilett (2004) used a historical data-based model, regression models and artificial neural network models to predict bus arrival time and found that ANN models outperformed the others in terms of prediction accuracy. van Hinsbergen et al. (2009) combined neural networks in a committee using Bayesian inference theory. An evidence factor was used as a stopping criterion during training and as a tool to select and combine different neural networks.

SVM models are a specific type of learning algorithm characterized by the capacity control of the decision function, the use of the kernel functions and the sparsity of solutions (Cristianini and Shawe-Taylor, 2000; Vapnik, 2013; Vapnik, 1999). Yu et al. (2006) suggested that SVM model is suitable for bus arrival time prediction based on historical data of bus arrival. But Chen et al. (2004) pointed out historical data had difficulty in handling dynamic traffic conditions owing to the lack of real-time data. Then Yu et al. (2010) developed a hybrid model combining a Kalman filtering method with a SVM model, which takes latest bus arrival information/real-time data into account.

However, neural networks and SVMs are complicated because of the large number of parameters needed to be adjusted. Additionally, these algorithms tends to overfit the data (Breiman et al., 1984). Highlighted interest focuses on the emerging type of machine learning technique in recent years, such as random forests, neural network ensembles, bagging and

boosting, and so on (Ghimire et al., 2010; Gislason et al., 2006; Sesnie et al., 2008; Steele, 2000). Ensemble learning algorithms work by running a "base learning algorithm" multiple times, and forming a vote out of the resulting hypotheses (Dietterich, 2002). Ensemble learning technique might have higher accuracy because the group of classifiers performs better than only one single classifier.

Random Forest (RF) model is constructed in a random vector of the data feature space (Breiman, 2001). RF models improve the accuracy of regression without a great increase in computation complexity. Additionally, these models can explain the importance of thousands of variables (Breiman, 2001; Iverson et al., 2008). RFs are efficient and accurate compared with other machine learning models; thus, they are widely applied in different fields (Cutler et al., 2007; Genuer et al., 2010; Yang et al., 2016). Generally, RFs have shorter calculation time and the problem of multicollinearity can be ignored. RF is not sensitive to outliers and remains robust despite missing data. Meanwhile, RF models can reduce overfitting (Breiman, 2001; Friedman and Meulman, 2003) because of the random selection in features and training samples.

RF has been applied in transport such as prediction of traffic flows (Hamner, 2010; Leshem and Ritov, 2007) and bus travel time prediction (Gal et al., 2015; Moreira, 2008). In Moreira's work, the travel time prediction is designed for planning purposes of mass transit companies in a relatively macroscopical aspect. That is, Moreira gave an application of three machine learning algorithms including RF, and travel times of whole trips for transit companies are predicted considering pay day impact, seasonality of the year, and so on. In general, Moreira's work focused on business transit in macroscopical aspect, which did not consider characteristics of buses and bus data, for example, bus dwell time, traffic conditions. Gal et al. (2015) combined queuing theory and machine learning to forecast the bus travel time, with the main concept of predicting travel time based on queuing theory and identifying outliers of the travel time by using machine learning. RF is one of the algorithms that was used in Gal's research for the detection of outliers in scheduled transportation. Gal et al. (2015) used the travel time of the preceding 1 bus as an estimate of the predicted one. Different from these researches, we propose a hybrid model, which combines RF and near neighbors, to forecast the travel time considering current traffic conditions both on current segment and next segments.

Near neighbors method and its extension are also used in prediction of bus travel time, namely KNN method. Main property of near neighbor regression is that the method needs few or no parameters, whose calibration will cost much time in computing. Chang et al. (2010) developed a model based on the nearest neighbor nonparametric regression using historical and current data from the AVL system. Different from the conventional nearest neighbors (KNN) method, where a certain number of samples are selected for prediction, we apply near neighbor method to calculate a weight for each sample in the selection of training set. The near neighbor method used in our article is the linear search (exact method) and compared with other search methods (e.g., K-d tree, KNN), linear search is simple and provides exact results, which can be applied with the help of cloud and parallel computation for faster computation.

The prediction of bus arrival time is usually considered in two ways. First, some researchers simultaneously consider bus running time and bus dwell time. Wall and Dailey (1999) used a combination of both AVL data and historical data to predict bus arrival time. A Kalman filter model is used to track vehicle location and predict bus travel time, where dwell time is not explicitly coped with as an independent variable. Chien et al. (2002) did not consider dwell time as input variables in their ANN model. Second, some researchers consider bus dwell time and running time separately (Jeong and Rilett, 2004; Shalaby and Farhan, 2003). In this article, bus running time and bus dwell time at stops are not separately considered and are not estimated. However, we combine them as the travel times of buses between bus stops. The bus dwell time is taken as a factor of bus travel time in our model.

In terms of model features, a lot of studies focus on the relation of bus travel time over the historical data. Some regression techniques predict the dependent variable by the formulation formed by a set of independent variables that affect travel time, which may include road and traffic conditions, weather, signals, intersections and driver characteristics (Bo et al., 2010; Patnaik et al., 2004). In our method, traffic conditions are mainly considered in model formulation owing to data availability of other factors. Compared with other factors such as weather, road conditions, and driver characteristics, traffic conditions gradually play an important role in affecting bus travel time, especially in congested cities.

The purpose of our work is to predict the bus travel time using the proposed method and analyze the performances in different cases, considering the current traffic conditions as the factors. The predicted travel time can be provided to travelers to help in decision making and can be used for bus operations. In addition, AVL systems and parallel and cooperative computing facilitate real-time prediction.

This paper aims to make two contributions. A new method for the prediction of bus travel time is proposed, that is, random forests based on the near neighbor (RFNN). RFNN involves random forests and the concept of near neighbor, in which a preselection process for training data set is posed to enhance the performance of random forests. Although the computation time of RFNN is longer, the results of RFNN show higher accuracies in mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE). RFNN makes it possible for bus operators and passengers to seek for more accurate predictions of bus arrival time. In addition, RFNN suits for large-scale data sets because of the extraction of similar samples and discard of unessential data in the entire data set. Incidentally, random forest is also applied to predict bus travel time in the manuscript, and the performance of RF is evaluated by the comparison with other methods. RF is rarely used in the field of prediction of bus travel time, especially in the prediction that considers traffic conditions as a factor.

The remainder of this article is organized as follows: Section 2 describes RFs and the proposed RFNN model. In Section 3, a numerical test with sensitivity analysis and state-of-the-art comparison are presented. Section 4 concludes the article and provides an outlook on future works.

2 METHOD

In this section, the concept of classic RFs is presented, and the proposed improvement algorithm (Random Forests based on near neighbors) is described.

2.1 Random forests

RF model is a kind of Classification and Regression Tree (CART) model and a type of ensemble learning algorithm. Considering the problem of overfitting, Breiman (2001) proposed the RF model that combines the results of multiple trees (forest) without a significant increase in computation complexity.

In decision tree (DT) learning, the term feature is commonly used, such as the independent variables in regression models. A feature is defined as the dimensions of a data set. For example, weather, road length, and traffic conditions may affect the prediction of bus travel time, which are also referred to as features in DT learning. In the procedure of training, each tree is built based on a random subset of features. For a specific data set, it has a set of features and one random subset of features is assigned to each built tree (also referred to as feature bagging). The reason for this process is the correlation of the trees. Specifically, if a few features are strong predictors for the target output, these features will be selected in many of the built trees, which cause them to become correlated (Ho, 2002). Typically, a third of the total number of features selected for each built tree is recommended. Meanwhile, each tree is trained with a random subset of the original data (Breiman, 2001). In the procedure of data selection, bootstrap sampling is employed, which enables the remaining unused subset to be used for calculating general errors. After the training (based on Information Gain) of the data set, RFs model returns the average output of all aggregations by voting. Different from the single decision tree, RFs is the combination of multiple decision trees. Each tree is an expert of classification or regression in a certain set of features. The final results of RF are obtained by the voting of all trees in RFs, which are superior to single classifier models (Liaw and Wiener, 2002).

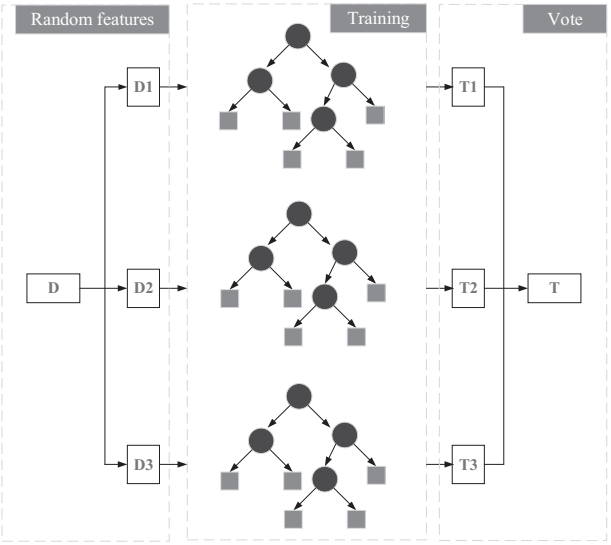


Fig. 1. Process of random forests.

For example, as depicted in Figure 1, assuming five features/independent variables ($d1, d2, \ldots, d5$) for each data, which jointly determine the value of the dependent variable (output), these five features are generated in a single tree for the regression in CART models. Nonetheless, RF model generates a forest with multiple trees where each tree generates with a random subset of the entire features ($D1, D2 \ldots$), assuming that each tree generates with three random features: $D1$ ($d1, d3, d4$), $D2$ ($d2, d3, d5$), and $D3$ ($d1, d2, d4$). Furthermore, each tree is trained based on Information Gain, and each tree majors in certain features (e.g., $D1$ experts in $d1, d3, d4$). That is, the concept of branching in RFs is to set the feature with the maximum information as the upper split feature (e.g., the upper gray circle of T1 in Figure 1). Then the prediction result is obtained according to the

vote of the forest ($T1$, $T2$, $T3$ stand for three different trees), where the average value of the outputs from all trees is commonly used for the final regression result (see Equation (1)). RF will divide the data set based on values of each feature and finally there might be several data at an end leaf. In regression, each tree outputs the mean value of those several data at one of the end leaves, and final prediction results are the mean of each tree. Note that each tree has the equivalent weight in voting.

$$H(x) = \frac{1}{T} \times \sum_{i=1}^{T} h_i(x) \tag{1}$$

where T is the number of trees in the forest and $h()$ stands for the prediction values of the ith tree.

RFs model requires a limit number of parameters (two main parameters): the number of trees in the forest (refer as n_{tree}) and the number of input variables/features used to generate each tree (refer as m_{try}). That is, n_{tree} represents the number of trees in the forests, whereas m_{try} represents the same number of features every tree in the forest contains. The influence of m_{try} means the strength of each tree and also the correlations between trees. A larger value indicates an increase in strength and correlation (Peters et al., 2008). The performance of prediction using RFs model is proved better by increasing the strength and decreasing the correlation, so the value of m_{try} is twofold (Ließ et al., 2012).

For regression problems the inventors recommend $D/3$ for m_{try} (D is the number of total features/input variables) as the default value. The default value of n_{tree} is 500, which was proven; however, it is not appropriate to obtain stable results (Grimm et al., 2008). Thus, we set $n_{\text{tree}} = 1{,}000$ in this article and the value changes in *sensitivity analysis* to figure out the best value of parameters.

2.2 RFs based on near neighbors

The scale of data might be a double-edged sword for arrival time prediction. Few data usually cause a lack of necessary information and hinder the capture of inner relations with its features. Nonetheless, the error rate of the prediction model usually increases in mass training data because the large scale of data may have many data that are not strongly relevant to predictions, which may negatively influence the prediction. To improve the accuracy of the entire prediction process in this article, we propose a hybridization method of the random forests model based on the near neighbors method, which imposes a process of preselection for the training set in RFs models. That is, RFNN contains two main procedures, where the first process involves selection of the training set for the RF model from the original data set, and the second process involves the training and regression procedure of RF model.

First, the preselection process of training set for RFs is based on the concept of "near neighbors," which can also be considered as the reorganization of original data, that is, the result of the preselection is taken as the training set of RF models. Training data set is necessary and essential for machine learning algorithm, but traffic data of public buses is vast thus contains many useless or noisy data for a specific prediction of bus travel time. Therefore, we attempt to identify high-quality data/samples for a certain prediction, which may improve the quality of training set and the prediction. The "high-quality" is measured by the similarity of the training set and predicted data, and the similarity is often numerically measured by the distance of the compared ones in aspect of data features. That is, similarity between samples and the reference sample is considered to be the quality of samples when compared with the reference sample, where higher similarity indicates higher quality. In bus travel time prediction, similar conditions of traffic usually lead to similar bus running conditions (e.g., running speed) and this similarity is captured by the preselection process of training sets. In other words, if we predict the bus running time for a specific traffic condition, we might learn a lot from those similar conditions, which entails the similarity and distance between the reference sample (data to be predicted) and other training samples. The distance between the reference sample and other samples are calculated as Equation (2), where the difference will be assigned a specific weight (see Figure 2) for selection (selection probability). The result of the selection (a set of samples) will be further set as the training set of RFs model. These "near neighbors" mean samples, which are close to the reference sample (in measure of distance). A commonly used distance metric for continuous variables is the Euclidean distance, whereas the Hamming distance or correlation coefficient is common for discrete variables. The Euclidean distance is chosen in this article for distance calculation according to the type of the collected data, as shown in Equation (2).

$$\text{distance}(X_0, X_1) = \sqrt{\sum_{i=1}^{|X_0|} (X_{0i} - X_{1i})^2} \tag{2}$$

where X_0 stands for the reference sample (the sample to be predicted) and X_1 represents the sample in the training set. X_{1i} represents the ith features of sample X_1, and the difference of samples is measured by the similarity of sample features/dimensions.

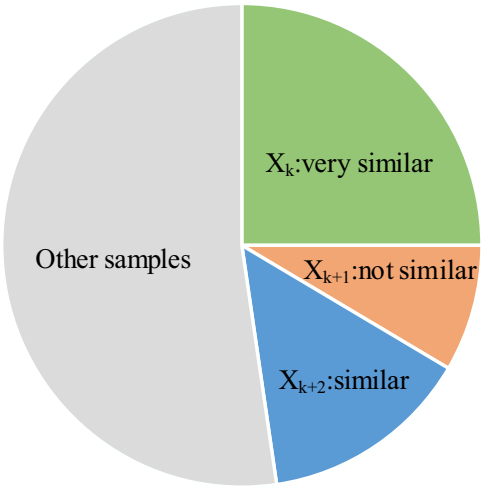


Fig. 2. Selection probability (Roulette) of each sample when comparing with the reference sample X_0. For example, X_k is a very similar sample compared with X_0 thus X_k has a higher choose probability than that of X_{k+1} and X_{k+2} in the selection of training set.

Second, the training samples for the RFs prediction model, which is a black-box regression process, are selected using the Roulette method, which is based on the selection probability (the obtained distance in the first process) of each sample. The training set of RF is selected from the original data set, which is based on a similarity-related probability. A higher selection probability is assigned to samples, which are similar to the data to be predicted (reference sample). Figure 2 shows the generation of the selection probability of samples. Results of the selection are the similar samples (similar traffic conditions) that have higher probability to be chosen in the training set from the vast sample set. However, the selected samples are not used for training of RF model directly. That is, a reselection process (bootstrap sampling) is used, which allows the remaining unused subsets to be used for calculating general errors, as commonly used in RF models. After the training of data sets, RFs model returns the average output of aggregations of trees in the forest.

The procedure of the proposed RFs based on near neighbors is depicted in Figure 3.

Bus travel time prediction between adjacent bus stops that considers current traffic conditions can be described as follows: bus travel time between bus stops (including bus running time and bus dwell time at stops) is assumed to have relations with the average bus dwell time of the current stop and the current traffic conditions on the predicted route segment and next segments. In Equation (3), $\hat{T}_{t,k}$ stands for the predicted value of bus travel time (from the start point of segment k to

```
1  Set num_data = number of initial dataset
2  Set num_training = number of training set for RFs
   model
3  //Initialization
4     Normalization of data
5     Divide the dataset into num_training for training set
6     and num_data-num_training for  test set
7  //Near neighbors (pre-selection)
8     For i = 1 to num_training
9        distance_i = Euclidean distance between data_i and
   the reference/predict sample
10       selection_probability_i =
   (max_distance-distance_i)/sum(max_distance-distance_i)
11       cumulative_probability_i =
   cumulative_probability_i-1 + selection_probability_i
12    End for
13    Select num_training samples based on Roulette
   method
14 //Random forests regression
15    Bootstrap selection for the num_training samples in
   former process
16    Calibration of parameters (ntree, mtry) in RFs
   model
17    RFNN_model = machine learning
18    prediction_result = average(prediction value of
   ntree trees based on RFNN_model)
```

Fig. 3. Procedure of the RFs based on near neighbors (RFNN).

the start point of segment $k+1$) on segment k at time t, and $c_{t,k+1}$ represents the current traffic conditions on segment $k+1$, which is the downstream segment of segment k. Two variables are used to measure traffic conditions: the average running speed $s_{t,k}$ and the speed variance $v_{t,k}$ of the segment. $\hat{T}_{t,k}$ is the combination of the running time on segment k and the dwell time $d_{t,k}$ at the end of segment k (bus stop dwell time) at time t. Current traffic conditions on segment k are measured by the average running speed and the variance of vehicles on segment k (refer to Equation (4)). $\hat{T}_{t,k}$ can be predicted by the set of variables in Equation (5), which are input variables in our machine learning algorithm: RFNN. Estimated values are used to replace the real values because of the availability of real-time data. For example, $s_{t,k}$ is not available at time t in bus data because it is rare that a bus sharply finishes the running on segment k at time t and then the average running speed on k can be calculated. Therefore, we use the average running speeds of the preceding buses, which have finished running on segment k, to approximately replace the average running speed at time t.

$$\hat{T}_{t,k} = f\left(d_{t,k}, c_{t,k}, c_{t,k+1}, \ldots\right) \tag{3}$$

$$c_{t,k} = h\left(s_{t,k}, v_{t,k}\right) \tag{4}$$

$$\hat{T}_{t,k} = g\left(d_{t,k}, s_{t,k}, v_{t,k}, s_{t,k+1}, v_{t,k+1} \ldots\right) \tag{5}$$

2.3 Model validation

The validation of the RF model can be measured in three indices: MAE, MAPE, and RMSE. The MAE measures the average magnitude of the errors in a set of forecasts, whereas the RMSE measures the average magnitude of the error. The RMSE gives a relatively high weight to large errors and is always larger or equal to the MAE. The greater difference between them, the greater the variance in the individual errors in the sample set. MAE and RMSE have the dimensions of prediction and observed values (second), whereas MAPE is dimensionless (%). The three indices are calculated for the validation data sets as follows:

$$\mathrm{MAE} = \frac{1}{n} \times \sum_{i=1}^{n} |f_i - y_i| \tag{6}$$

$$\mathrm{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \frac{|f_i - y_i|}{y_i} \times 100\% \tag{7}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n-1} \times \sum_{i=1}^{n} (f_i - y_i)^2} \tag{8}$$

where f_i is the prediction result, y_i is the observed value (real value), and n is the number of samples.

3 NUMERICAL TEST

3.1 Data collection and analysis

The model for bus arrival time prediction has been tested with the data sets of bus routes 232 and 249 in Shenyang, which is the capital of Liaoning Province in China. The two bus routes extend from the suburb to the center of Shenyang without timetables at each bus stop. Bus route 232 has 19 bus stops and the total distance extends 10.7 km. Meanwhile, the total travel time from origin to destination is approximately 60 minutes, with the bus frequency of 2.5 minutes. Bus route 249 has a length of about 15 km with the bus frequency of about 7 minutes and 27 bus stops. In aspect of bus running speeds, bus routes 232 and 249 have the average running speed of 15.6 km/hour and 14.9 km/hour, respectively, which indicates similar traffic conditions on the two bus routes. Seventeen out of the 18 route segments (divided by the bus stops) are set as time points for arrival time prediction in Figure 3. Similarly, 25 of the 26 route segments are set as the time points for bus arrival time prediction. Because the proposed method predicts bus travel times based on the running conditions on the next segment, the last segment of each bus route is removed from the prediction. The data set is collected on 23/02/2016–25/02/2016 (06:30–19:30) with 15,743 original data of bus arrivals for bus route 232 and 8,257 data for route 249 from the automotive vehicle location (AVL) system. Data of bus arrivals and departures at bus stops are obtained after map matching and cleaning of abnormal data. The remainder of the data set (14,182 data for bus route 232 and 7,623 for bus route 249) is divided into two subsets: 80% for training and 20% for testing. Table 1 lists the descriptive statistics of the data set and Figure 4 depicts the direction of bus routes 232 and 249, where only one direction of bus routes is selected for the case.

As depicted in Figure 5, Bus a is the bus that required predictions, whereas Bus b, …, e are the preceding buses of Bus a. The prediction process enables us to figure out the travel time (including running time and dwell time) of Bus a on Segment 1. We can obtain the running information (average speed and speed variance) of buses from the AVL systems, both on Segment 1 and Segment 2. The running information is updated when the preceding buses leave a stop. That is, the traffic conditions of Segment 1 are updated and measured with the arrival data of Bus c, which finishes travel between bus stops and just leaves for the next stop. Considering Figure 5 as an example, the bus running information is updated at time t. Arrival information of Bus c is the latest information that reflects the current traffic conditions However, the running information of Buses d and e can also be used to measure the traffic conditions on Segment 1. Because Bus c has a distance with Bus d, the main difference between the running information of the two buses is the information loss of current traffic during the "distance." In practice, a shorter interval of data updating will result in a better estimation of current traffic conditions and it is set as a 1-minute interval for updating input data considering practical application. As shown in Figure 5, we consider Bus c and Bus d as the preceding two buses of Bus a because Bus b has not finished travel on Segment 1. Meanwhile, Bus e is regarded as the preceding bus of Bus a on the next (downstream) segment.

3.2 Results

Generally there are two main parameters while using RFs: m_{try} and n_{tree}. However, the best values of m_{try} and n_{tree} for our prediction problem are unknown. Thus a searching of the parameter values should be conducted first, which aims to identify suitable value of parameters (m_{try} and n_{tree}) to predict unknown data accurately. We set the value of the two parameters to 1,000 (as previously mentioned) and $D/3$ (default value), respectively. In the next section, we discuss the sensitivity of the results when parameters' values change.

Table 1
Descriptive statistics for the collected data: route segment number, DataSize, road length, min, max, mean, and standard deviation of bus running time on each route segment. Left: 232; right: 249

	DataSize		*Length [m]*		*Min [s]*		*Max [s]*		*Mean [s]*		*SD [s]*	
Seg	*232*	*249*	*232*	*249*	*232*	*249*	*232*	*249*	*232*	*249*	*232*	*249*
1	861	310	596	320	59	89	153	142	85	117	13.28	10.87
2	769	290	703	900	101	90	271	245	143	187	22.43	28.40
3	858	309	350	488	47	87	137	149	68	116	10.98	17.41
4	851	298	980	458	107	59	250	170	177	115	31.84	19.45
5	874	296	1,000	740	163	200	657	452	306	325	78.48	45.32
6	787	310	270	517	44	100	127	243	66	174	9.82	28.90
7	894	304	605	922	96	168	283	321	135	232	21.48	30.69
8	797	306	615	615	75	115	174	228	106	157	20.22	27.92
9	794	311	670	555	162	152	514	332	282	260	61.81	36.57
10	866	324	500	1,100	107	190	304	570	164	299	21.93	84.47
11	816	288	520	607	85	145	217	347	128	239	19.91	44.05
12	858	304	1,000	385	159	73	395	147	237	106	43.55	22.49
13	862	314	485	470	84	127	211	252	124	204	21.18	59.60
14	824	324	353	285	129	124	293	256	189	185	31.27	59.53
15	870	298	322	680	77	119	172	302	107	222	13.75	37.76
16	794	302	845	740	139	134	423	267	217	200	57.86	25.51
17	807	326	510	728	89	121	220	475	119	200	15.91	54.65
18		322		399		104		387		236		67.35
19		283		680		135		293		221		52.36
20		285		228		73		204		143		36.25
21		307		920		180		332		224		45.09
22		322		445		42		120		75		31.10
23		280		357		52		208		95		30.06
24		316		650		67		136		98		24.33
25		294		760		126		290		201		35.27

The candidate input variables in the model could be divided into three aspects: average bus dwell time at the stop, the running information/conditions of the buses on the current route segment and the next (downstream) route segment. In other words, the condition (e.g., average speed and speed variance) of the current route segment reflects the current traffic conditions (e.g., congestions) and specific route segment conditions (e.g., condition of poor road pavements). For instance, one route segment with a poor pavement condition usually has slower speeds for buses. Furthermore, the condition of the next route segment could also affect the running buses on the current segment. That is, congestions on the next route segment could play a negative role in the interference on the current route segments because of traffic waves. Therefore, the traffic conditions of the current segment and only one downstream (the next) segment are selected in the case study. Traffic conditions are measured by the average running speed and the variance of average speeds, which are collected from the data of preceding buses in the AVL system. In addition, the bus dwell time at stop k is calculated and set as the average value of dwell time at stop k in the time interval of one hour. Although bus dwell time can be estimated with APC data, we use average values for our predictions owing to the lack of APC data. Specific input variables are listed in Table 2.

In Table 2, BDT stands for the average bus dwell time at the bus stop in an hour. To illustrate relative magnitudes of bus dwell time at different stops, bus dwell time is measured in proportion. That is, an index of bus dwell time at a stop is employed to represent the proportion of dwell time at a bus stop against the total dwell time of stops. It is easy to understand and calculated as

$$IBDT = \frac{1}{M} \sum_{m \in M} \frac{BDT_{ms}}{\sum_{s \in S} BDT_{ms}} \tag{9}$$

where $IBDT$ represents the index of bus dwell time at a bus stop, M is the set of buses, and BDT_{ms} stands for the bus dwell time of bus m at stop s. Figure 6 illustrates the average bus dwell time at stops among the two routes. Compared with bus route 249, route 232 has much more stops that have longer bus dwell time than the average

Fig. 4. Bus routes 232 and 249. Left for bus route 232; right: 249. The direction of the bus routes discussed in this article is from the top to the bottom.

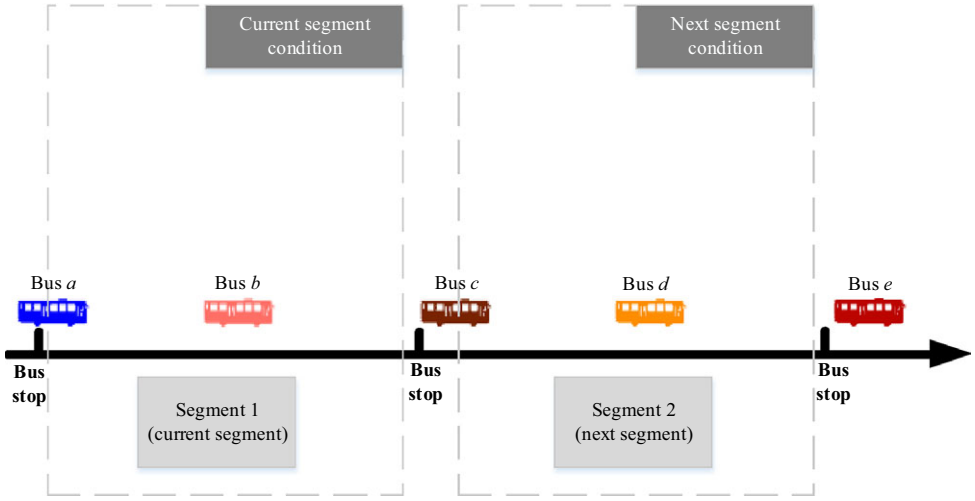


Fig. 5. Explanation of bus arrival time prediction process at time *t*.

values, which indicates larger passenger flows at these bus stops.

SC2 indicates speed_current_2buses, that is, the average of the average running speed of the preceding two buses on the current segment, whereas VC2 represents the speed variance of the two buses on the current segment and SN1 denotes speed_next_1bus on the next segment. All these parameters are set as the input variables in the basic scenario (S0). The bus travel time is tested on the test data set (20% of the whole data) 10 times, and each time of prediction will output different values owing to the preselection based on near neighbors and bootstrap selection of training set in RF. In Figures 7 and 8, three lines are depicted to represent

Table 2
Input variables

Input variable			*Symbol*
Bus dwell time at stop			BDT
Running condition of the buses on the current route segment	Preceding 1 bus	Average speed	SC1
	Preceding 2 buses	Average speed	SC2
		Speed variance	VC2
	Preceding 3 buses	Average speed	SC3
		Speed variance	VC3
Running condition of the buses on the next route segment	Preceding 1 bus	Average speed	SN1
	Preceding 2 buses	Average speed	SN2
	Preceding 3 buses	Speed variance	VN2
		Average speed	SN3
		Speed variance	VN3

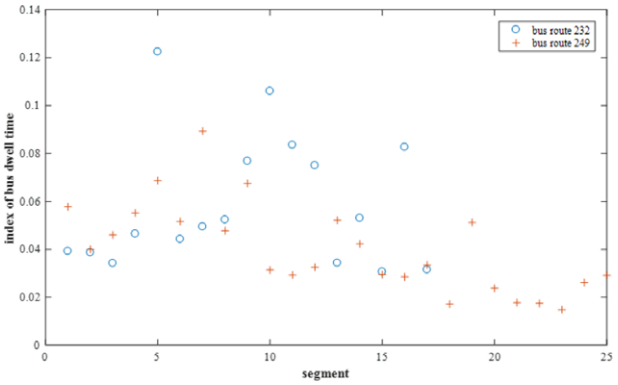


Fig. 6. Index of bus dwell time at segments.

the absolute deviations of 50 seconds. From the figures, it can be found that RFNN performs well because most prediction has an absolute deviation smaller than 50 seconds in Figure 7. As shown in Figure 8, the results show larger deviations of the prediction for bus route 249 than the deviations of the prediction for bus route 232. Many predictions of RFNN provide considerably accurate results because of the proper detection of similar traffic conditions. Note that in RFNN method, similarity is primarily measured by traffic conditions on segments; traffic conditions on a certain segment are assumed to be similar with the traffic conditions on other segments. That is, the similar samples in our pre-

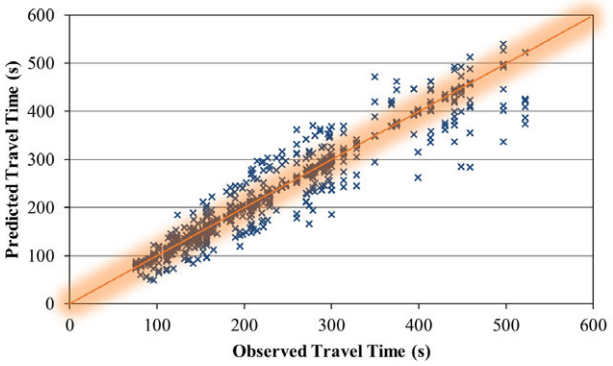


Fig. 7. Prediction results of S0 compared with the observed data of bus route 232. One hundred observed samples are randomly selected from the test data set and each observed sample is predicted by RFNN for 10 times.

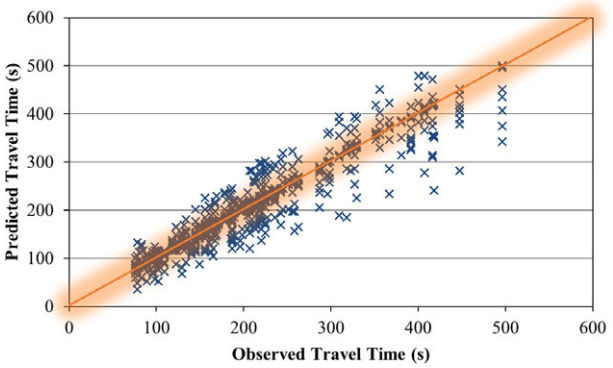


Fig. 8. Prediction results of S0 compared with the observed data of bus route 249. One hundred observed samples are randomly selected from the test data set and each observed sample is predicted by RFNN for 10 times.

selection process are selected from the data of not only a specific segment, but also other segments with similar traffic conditions. This assumption is motivated by the fact that similar traffic conditions on different segments usually lead to similar running conditions (e.g., running speed) and the quality of detecting the right current traffic conditions directly affects prediction performance. Figure 8 shows weaker identification of current traffic conditions as a result of longer bus frequencies and longer data updating owing to fewer buses, which is employed for estimating current traffic conditions. The results also imply that a significant amount of data will support the identification of similar conditions (data set of route 232 is almost twice the size of the data set of route 249).

In this case, the frequency of bus routes 232 and 249 is 2.5 minutes and 7 minutes, respectively. Assuming that the condition/information of the preceding buses

Table 3
Results of the 10 scenarios with 10 runs. Left for bus route 232 and right for 249

			MAE [s]		*MAPE [%]*		*RMSE [s]*	
Scenario	*Description*	*Input variable*	*232*	*249*	*232*	*249*	*232*	*249*
S0	Using all input variables: average speed and speed variance of the preceding 1, 2, and 3 buses on the current segment and the next segment	All	13.65	13.79	6.90	7.60	26.37	29.04
S11	Using the average speed and speed variance of the preceding 1 bus on the current segment and that of the preceding 1 bus on the next segment	BDT,SC1,SN1	15.24	15.06	7.85	8.63	30.96	30.70
S12	Current segment: 1 bus; next segment: 2 buses	BDT,SC1,SN2, VN2	15.06	14.50	7.69	8.15	32.23	30.68
S13	Current segment: 1 bus; next segment: 3 buses	BDT,SC1,SN3, VN3	15.48	14.13	7.85	7.91	32.66	29.49
S21	Current segment: 2 buses; next segment: 1 bus	BDT,SC2,VC2, SN1	19.28	17.93	10.90	11.21	38.77	36.44
S22	Current segment: 2 buses; next segment: 2 buses	BDT,SC2,VC2,SN2, VN2	17.73	17.19	10.10	10.88	37.00	35.82
S23	Current segment: 2 buses; next segment: 3 buses	BDT,SC2,VC2,SN3, VN3	17.77	17.39	10.16	10.90	33.37	32.24
S31	Current segment: 3 buses; next segment: 1 bus	BDT,SC3,VC3, SN1	22.65	22.01	13.41	14.29	44.27	42.86
S32	Current segment: 3 buses; next segment: 2 buses	BDT,SC3,VC3,SN2, VN2	21.07	20.67	12.51	13.33	39.32	38.09
S33	Current segment: 3 buses; next segment: 3 buses	BDT,SC3,VC3,SN3, VN3	21.30	20.49	12.71	13.26	38.11	36.33

reflects the condition of traffic, we use nine additional scenarios to obtain information (average speed and speed variance) about how many preceding buses could obtain a better prediction accuracy. It seems that the preceding one bus usually has a stronger relation with the current bus (need to be predicted), whereas the average running time of the preceding two buses or three buses has a weaker relation due to a long time interval (refer to the bus frequency/headway) between adjacent buses, which may result in a low-accuracy estimation of the current traffic conditions. From Table 3, the results of bus route 249 with frequency of 7 minutes show a significant distinct trend in which a longer headway (lower frequency) between adjacent buses tends to yield lower accuracy compared with the results of bus route 232. The loss in accuracy is attributed to the data collection and updating method in our article. That is, the interval of the data update and bus headway (frequency) have a combined influence on data availability. A longer headway results in fewer buses on the entire bus route and has a smaller number of bus arrival information during a determined time range, which contributes to a lack of reflection of current traffic conditions. Meanwhile, the availability of

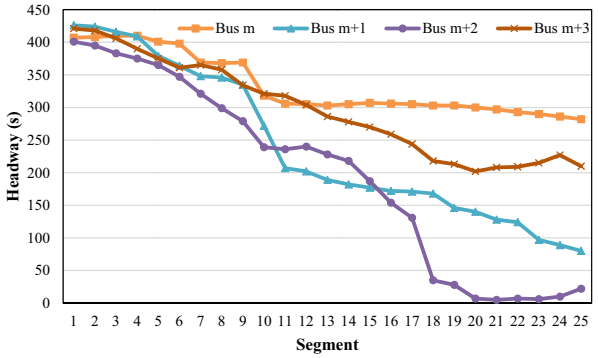


Fig. 9. Headway of buses for bus route 249 at different segment. Bus m is the first bus, and $m + 1$ etc. are the following buses. For example, plot of bus $m + 2$ is the headway between bus $m + 2$ and its preceding bus (bus $m + 1$). Furthermore, headway is calculated by the gap between the departure times of two sequential buses at stops.

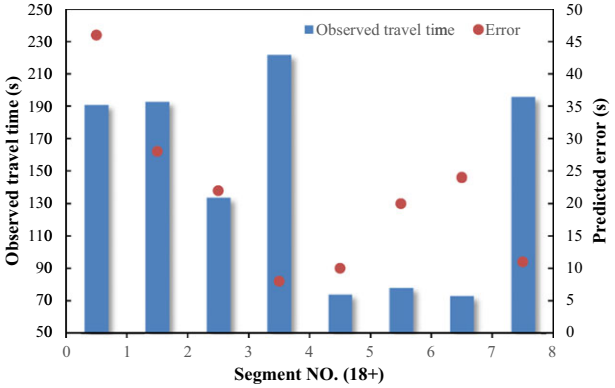


Fig. 10. Observed travel time and predicted error of segment 18–25 for bus $m + 2$. The prediction of RFNN is conducted for 10 times and the average of predicted values are used to calculate errors.

real-time data caused by the interval of the data update is distinct.

Nonetheless, if the running information of the preceding one bus is emphasized, we may get a low-accuracy estimation when unexpected events act on the preceding one bus (e.g., car accidents). What about the average running conditions of the preceding 2 or 3 buses? Therefore, 10 scenarios (with one basic scenario S0) are set to describe the cases in which the running conditions of different numbers of preceding buses are employed. The input variables and results of the 10 scenarios with 20 runs are listed in Table 3. In Table 3, S0 is the basic scenario, and all input variables listed in Table 2 are applied into the RFNN model, whereas other variables are different scenarios with diverse input variables. For example, S23 indicates that the input variables are (1) the average bus dwell time at stop on the current segment, (2) the average running conditions of the preceding two buses on the current segment, and (3) the average conditions of the preceding three buses on the next segment in RFNN model. The input variables can also be described as average bus dwell time (BDT), the average running speed of the preceding two buses (SC2) and the speed variances (VC2) on the current segment; the average running speeds of the preceding three buses (SN3) and the speed variances (VN3) on the next segment.

To evaluate the performance of the proposed method when addressing bus bunching, which indicates large gaps in headway, we detect and choose a real case from bus route 249. A bus bunching occurs between segment 18 and segment 25, especially for bus $m + 1$ and $m + 2$ in Figure 9. Serious bus bunching occurs between bus $m + 1$ and $m + 2$, whereas bus $m + 3$ weakens this deterioration. Therefore, the data of bus $m + 2$ is removed from the training set and added into the testing data set to test the performance of RFNN when facing up with bus bunching.

The results of bus $m + 2$ are depicted in Figure 10, where the predicted values of bus $m + 2$ from segment 18 to 25 exceed the observed values. The reason for these positive errors is the short headway between bus $m + 1$ and bus $m + 2$, in which generates fewer waiting passengers and shorter bus dwell times. However, the short headway (bus bunching) will prevent the real-time data updating for the predicted bus (bus $m + 2$). That is, the real-time data of bus $m + 1$ is not updated in time because it might not finish the running on the current segment. Therefore, the values of input variables for bus $m + 2$ are also not updated and the predicted values are larger than the observed values, which underestimates the changes in the number of waiting passengers owing to shorter headway, especially for the bus stops with a larger variety in waiting passengers.

Traffic congestions usually cause larger variances of bus travel time. Accuracy of the prediction method in the condition of traffic congestion is essential to evaluate the performance. Figure 11 illustrates the performance of RFNN for the prediction of buses during morning peak hours on route 232. The four bus stops (segments) with largest average bus dwell times on bus route 232 are depicted, that is segments 5, 10, 11, and 16. Segments 11 and 16 of bus route 232 have larger prediction errors (with the MAPEs of 8% – 9%). Traffic conditions (e.g., congestions) contribute to the larger error rates partly, which can be also found in segments 5 and 10 (with MAPEs about 7.5 %). Another reason for the higher error rates of segments 11 and 16 than segments 5 and 10 is the similarity between segments

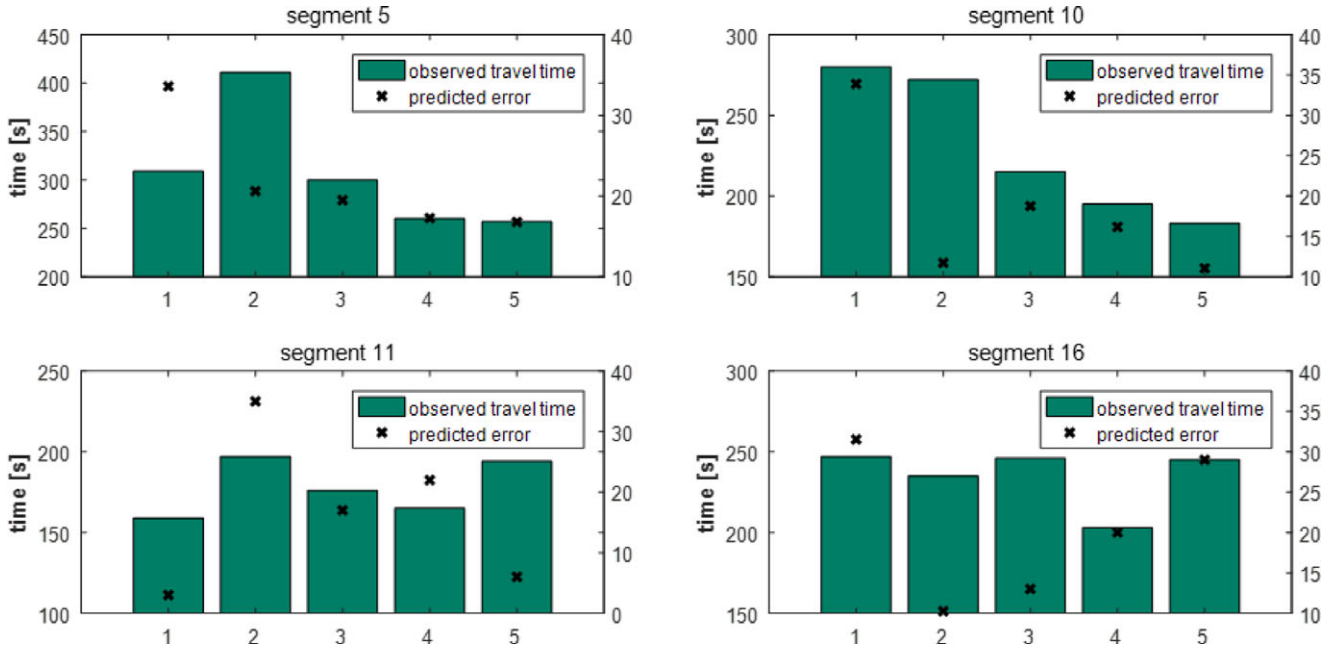


Fig. 11. Observed travel time and predicted error of segments 5, 10, 11, and 16 (four segments with the longest bus dwell times) for bus route 232. The prediction of RFNN is conducted for 10 times and the average of predicted values are used to calculate absolute errors. Five data of each segment are extracted from the testing data set.

11 and 16. Segments 11 and 16 have similar dwell times (mentioned in Figure 6) at stops, and the running speeds of buses during congestions are similar for all buses. The two reasons lead to a negative effect on the detection and selection of similar samples in RFNN. However, the results also indicate that RFNN has an acceptable prediction accuracy during peak hours at bus stops with large dwell times.

From the results in Table 3, we find that the accuracy of bus route 249 (MAPE) is lower than the accuracy of bus route 232 in all scenarios, which is resulting from the higher frequency of bus route 249 and the weaker reflection of the current traffic conditions due to the data updating method in this article. Although certain MAE and RMSE of route 249 are smaller than the MAE and RMSE of 232, note that the MAE and RMSE is usually related to the characteristics of the data. In these two cases, bus route 249 has more segments with a lower travel time in each segment, which causes smaller MAE and RMSE. The best performance of RFNN appears when all input variables are selected in the model training, which implies the strength of RF in discovering inner relations among many factors/features. Moreover, the prediction results of {S11, S12, S13} are better than the prediction results of {S21, S22, S23}, whereas {S21, S22, S23} is better than {S31, S32, S33}. This trend implies that less average running information on current segments are of significance for MAE, MAPE, and RMSE because traffic conditions change rapidly and an average of several preceding buses will weaken the reflection and evaluation of current traffic conditions. The running condition of the preceding one bus on current segments could reflect the traffic conditions better compared with the average conditions of two or three buses, at least in these two cases. The problem is the information delay (see Section 3.1, Data collection and analysis), which will result in a worse reflection of traffic conditions. Therefore, scenarios of one bus on the current segment show better performance owing to its up-to-date reflection of traffic conditions, whereas average of two or three buses on current segment weakens the effect of current traffic conditions. Nevertheless, fewer buses on current segments might not always perform well in all cases. For example, if the preceding one bus breaks down during bus operation, the current traffic conditions cannot be implied by this bus, and the average of the preceding two or more buses will weaken the negative effects of the breakdown.

In terms of the running conditions on next segments, there seems a trend that more buses on next segments could enhance the accuracy of predictions. For example, the trend in {S31, S32, S33}, implies that in this case, the average running conditions of more preceding buses shows the common traffic conditions of the next segment, which probably have a long-term interference on the buses that run on the current segment.

Table 4
Prediction results against different n_{tree} and m_{try} with 10 runs. Left for bus route 232 and right for bus route 249

		MAE [s]		*MAPE [%]*		*RMSE [s]*	
n_{tree}	m_{try}	*232*	*249*	*232*	*249*	*232*	*249*
500	1	19.58	19.05	10.53	11.33	35.29	33.85
500	2	17.12	16.12	9.08	9.45	31.87	29.74
500	3	15.73	14.59	8.07	8.27	30.07	27.81
500	4	15.44	14.41	7.59	7.88	29.68	27.00
1,000	1	19.54	18.37	10.63	10.90	35.10	33.15
1,000	2	16.59	15.61	8.90	9.31	31.22	29.24
1,000	3	15.34	14.68	7.93	8.41	29.49	27.68
1,000	4	13.65	13.79	6.90	7.60	26.37	29.04
1,500	1	19.62	18.32	10.54	10.94	35.42	32.80
1,500	2	15.97	15.42	8.65	9.25	30.48	29.04
1,500	3	15.40	14.36	7.95	8.26	29.43	27.17
1,500	4	15.27	14.35	7.58	7.82	28.67	26.98

The performance of S0 is better than most of the scenarios with acceptable accuracy (MAE, MAPE, and RMSE). Therefore, if the interrelations among input variables are unknown, the use of all these variables in RFNN is sometimes reasonable. Actually, RFNN and RF models generate a forest with many trees, and each tree performs well (experts) in certain features (input variables). The results of RF models consider all of these trees and their own skilled features. Thus, the results of using all input variables without variables screening can be accepted as good quality of results.

In summary, the results for bus routes 232 and 249 have a similar trend in aspect of the input variables and prediction accuracy. Because the traffic conditions of the two bus routes are similar because the average running speeds of buses on the routes have a small difference (0.7 km/h), the main difference of the prediction accuracy between these two bus routes is caused by the availability of real-time data resulting from diverse headways (bus frequencies). MAPE of bus route 249 tends to be higher than the MAPE of route 232 in most scenarios because of the lack of real-time traffic information. RFNN seems to show a better performance in accuracy when the current traffic conditions are updated with new running data of buses.

3.3 Sensitivity analysis

We perform a sensitivity analysis of the two main parameters (n_{tree} and m_{try}) in RF and RFNN models, which, as mentioned earlier, may have a significant effect on the performance of RFNN. In the basic scenario S0, n_{tree} and m_{try} are set to 1,000 and 4, respectively. In this section, we attempt to evaluate the interference of the parameters on the accuracy of the prediction.

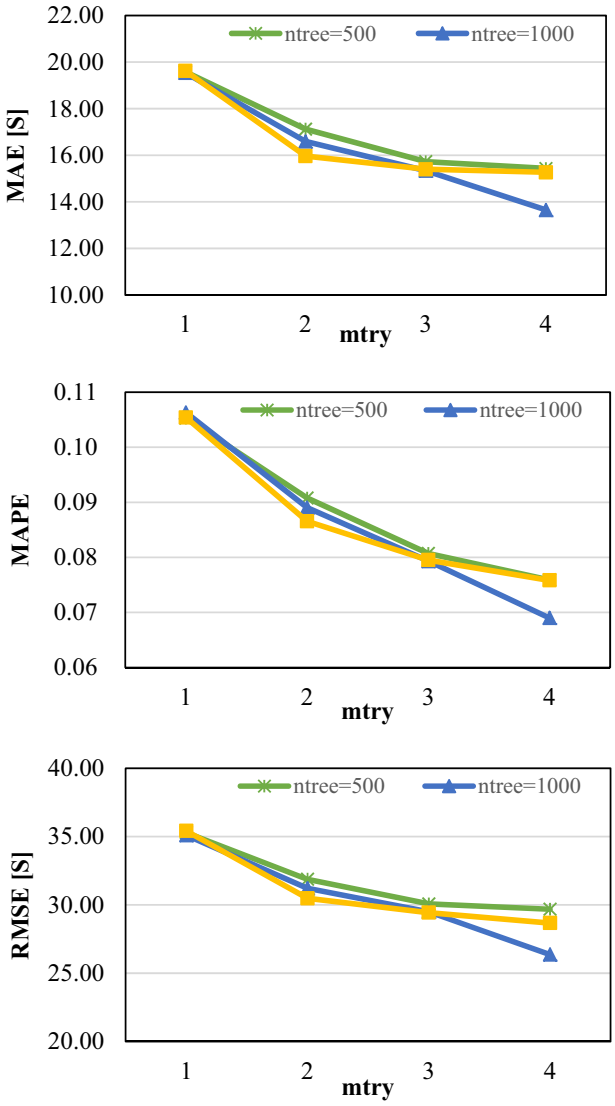


Fig. 12. MAE, MAPE, and RMSE of prediction results of bus route 232 against n_{tree} and m_{try}

The results show that a high value of m_{try} leads to better performance of RFNN especially for MAPE (see Table 4 and Figure 12), because m_{try} determines the strength of each individual tree and a large m_{try} increases this strength (Peters et al., 2008). The best results of MAPE and MAE are obtained when $n_{tree} = 1{,}000$ and $m_{try} = 4$, where $m_{try} = 4 \approx D/3$ (default value). Furthermore, the value of n_{tree} seems to have a weak interference on the prediction results, with the largest deviation of 1.78 seconds, 0.69%, and 3.32 seconds on MAE, MAPE, and RMSE, respectively, for bus routes 232 and 249. The smallest value of RMSE of

Table 5
Results of travel time prediction of five methods. Left for bus route 232, right for 249. Computation is conducted on the computer with dual-core 3.2 GHz processor and 4 GB RAM, and the computation time of RF and RFNN is collected for one time computation. Indeed, the two methods are computed 10 times

	MAE [s]		*MAPE [%]*		*RMSE [s]*		*Computation time [s]*	
Method	*232*	*249*	*232*	*249*	*232*	*249*	*232*	*249*
LR	31.30	31.61	16.41	18.05	46.77	44.93	104	42
KNN	32.66	31.35	17.33	17.77	48.47	45.24	25,216	3,715
SVM	21.09	21.28	11.16	12.37	31.24	30.33	7,112	2,405
RF	16.13	16.41	8.24	8.76	30.61	30.35	1,241	634
RFNN	13.65	13.77	6.90	7.58	26.37	29.01	44,301	6,286

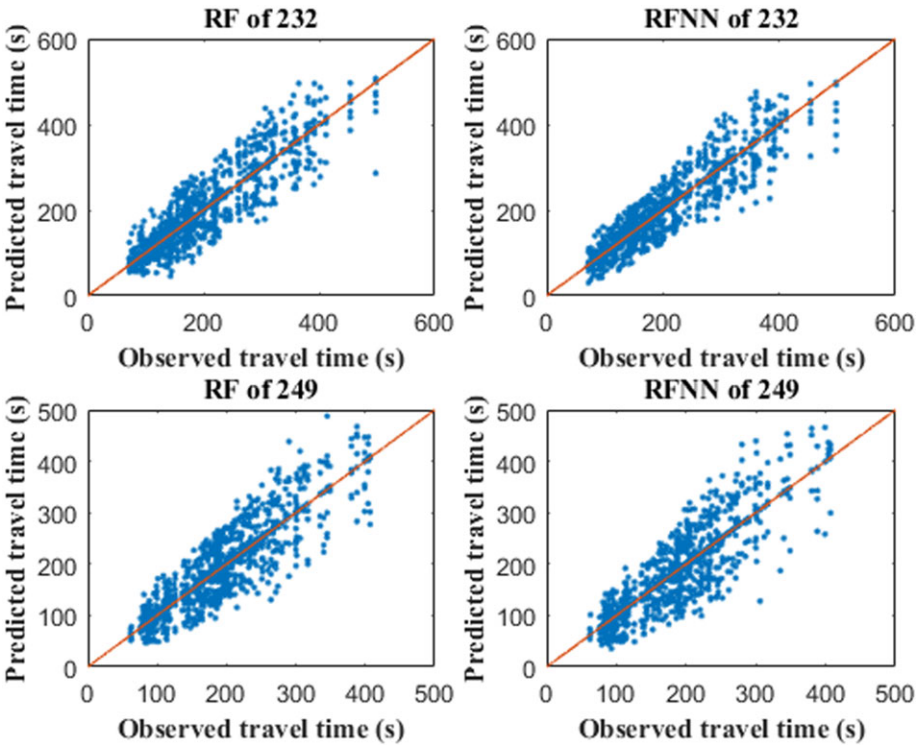


Fig. 13. Prediction results of S0 compared with the observed data. Two hundred observed samples are randomly selected from the test data set and each observed sample is predicted by RFNN and RF for 10 times.

bus route 249 occurs in $n_{\text{tree}} = 1{,}500$, but in general, the setting of parameters in S0 performs well. The results also imply that parameter calibration is necessary for better performance of the prediction in each case.

3.4 Comparison

The results of RFNN are compared in this section with other four main methods: Linear regression (LR); KNN; SVM; and classic random forest (RF). The results are listed in Table 5. These four methods have their strengths in prediction. LR is a classic and prevalent method for prediction due to its ability of analysis. KNN is a well-known nonparameter regression method that does not require the calibration of parameters. In machine learning methods, SVM has been proven to achieve higher accuracy of regression, especially for small-scale of data. Furthermore, the proposed RFNN is also compared with these typical methods, as well as RF method.

For a fair comparison among different methods, the training set and test set are randomly selected from the original data set but are set the same for the five methods. In addition, the results of RF and RFNN change

in each prediction because of the randomness in selecting training sets. Thus, both methods are computed 10 times to evaluate their performance. As shown in the table, RFNN has the highest accuracy, whereas LR model performs worst probably because of the interrelations of the input variables. The input variables (independent variables) in LR are set the same as RFNN and other methods, which indicates that all input variables in Table 2 are employed without considering the problem of interrelations among the independent variables. All input variables are selected because the selection of proper input variables or independent variables for different cases every time is complicated. The performance of each method in obtaining the inner relations among a number of input variables for high-quality prediction can be evaluated from this setting. Other methods are also set to use all variables in Table 2 for fair comparison in accuracy and computation time. The parameter k in KNN model in this article varies from 1 to 4 and is finally set to 3 according to the comparison of accuracy. When $k = 1$ or 2, the accuracy (MAE, MAPE, and RMSE) of KNN is worse than the accuracy when $k = 3$. Therefore, we adopt the best results ($k = 3$) for the further comparison with other methods. The computation time of KNN is considerable so that values of k are not larger than four. The values of main parameters in SVM are obtained using a 10-fold cross validation (McLachlan et al., 2005) and a grid search, where the main parameters are set to $c = 2^5, \varepsilon = 0.2$ using the core function of epsilon-SVR for regression.

For a fair comparison with RFNN, RF model is established with the same parameter values as RFNN, that is, $n_{\text{tree}} = 1,000, m_{\text{try}} = 4$. RFNN has better prediction results than classic RF model owing to the preselection process in RFNN because their values of parameters and input variables are equivalent. The preselection based on near neighbors method for the training set has a positive effect on the prediction accuracy. As shown in Figure 13, RFNN performs better in larger scale of data set (bus route 232) because more data could provide more similar samples for strengthening the process of preselection. For bus route 249, the better performance of RFNN is not clear (similar RMSE with RF) in the figure due to the lack of enough similar and helpful data. However, the preselection in RFNN also supports for the accuracy in MAE and MAPE.

In terms of CPU time, LR has the shortest running time, whereas the computation time of RFNN and KNN model is significantly longer owing to the time-consuming process of measuring distance/similarity in these two methods. The other two black-box methods (RF and SVM) have respectively shorter times than KNN and RFNN. Although RFNN needs a high occupation of CPU time, higher accuracy is possible. With the help of parallel computing and cloud computing, RFNN model can be applied to the real-time prediction of bus travel or arrival time based on the AVL system. Although LR model is direct and can be analyzed compared with black-box methods, the accuracy of LR is limited by the selection of independent variables in the model, and the inner relations between independent variables cannot be neglected. In contrary, RFNN model can handle cases with a larger number of input variables. With the development of ITS and ATIS systems, additional types of data and influence factors can be collected for bus travel time prediction, and machine learning methods (e.g., RF, RFNN) could also perform well with vast potential factors and provide high-quality predictions.

4 CONCLUSIONS

This article focuses on the prediction of bus travel time because it is vital to helping passengers decide departure times to bus stops and reducing anxiety of waiting passengers. RF model has a good performance in nonlinear regression and experts in coping with high-dimension variables or data. Few studies discuss the application of RFs in the prediction of bus travel time. Therefore, we propose a RF-based method that combines RF and near neighbors method. A preselection process of training set is conducted to extract similar samples from the entire data set, which can reduce computation time and eliminate negative effects of noisy and useless data, especially for a large-scale data set. Running data of buses, which contains location and time, on bus routes 232 and 249 in Shenyang are collected and extracted from the GPS data of buses. Then, the bus travel time (running time and dwell time) between adjacent bus stops is calculated based on these data and subsequently employed for model training in the proposed RFNN. The information about bus travel time is applied to measure current traffic conditions for the consideration of data availability and privacy. We consider the traffic conditions of both the current segment and the next segment as the input variables of the RFNN model. In a numerical test, we analyze the influence of the main parameters in RFNN and the effectiveness of the current traffic condition. To be specific, we discuss how many buses' running information can better reflect the current traffic conditions and improve the prediction accuracy. It seems that this number changes in different cases because of bus frequency, generally, availability of real-time data. Finally, we compare the RFNN method with four typical methods in bus travel time prediction. The results show that RFNN has a better performance in accuracy but not

computation time. With the help of parallel computing technology and better performance of computers, the long computation time does not pose a problem considering the high accuracy in prediction. More transit data can be collected from various equipment nowadays, and machine learning methods (including black-box methods) might be more proper to detect the relations between vast factors and bus travel time than conventional methods, although the analysis of these relations is not always feasible. The method proposed in the article can be applied to the prediction of bus travel time and can be easily extended to estimate bus arrival time at each bus stop based on current traffic conditions.

Our method can also be supported by APC data to enhance the estimation of bus dwell time at bus stops. Combined with the technology of parallel computing and cloud computing, reducing computation time to a low level and providing real-time prediction by setting a shorter interval of data updating is possible. In this article, only bus running data are employed for prediction. Further study will consider factors such as weather and numbers of waiting passengers. The average value of running speed can be changed to a weighted average speed where closer buses have larger weights.

ACKNOWLEDGMENTS

This research was supported in Natural Science Foundation of China 71571026 and 51578112, Liaoning Excellent Talents in University LR2015008 and the Fundamental Research Funds for the Central Universities (YWF-16-BJ-J-40 and DUT16YQ104).

REFERENCES

Adeli, H. (2001), Neural networks in civil engineering: 1989–2000, *Computer-Aided Civil and Infrastructure Engineering*, **16**(2), 126–42.

Adeli, H. & Hung, S.-L. (1994), *Machine Learning: Neural Networks, Genetic Algorithms, and Fuzzy Systems*, John Wiley & Sons, Inc., New York, NY.

Adeli, H. & Yeh, C. (1990), Neural network learning in engineering design, in *Paper presented at the Proceedings of the International Neural Network Conference*, Paris, France. Vol. 1, 412–15.

Al-Deek, H., D'Angelo, M. P. & Wang, M. (1998), Travel time prediction with non-linear time series, in *Paper presented at the Fifth International Conference on Applications of Advanced Technologies in Transportation Engineering*, Newport Beach, CA, 317–42.

Ben-Akiva, M. E. & Lerman, S. R. (1985), *Discrete Choice Analysis: Theory and Application to Travel Demand*, vol. 9, MIT Press, Cambridge, MA.

Bo, Y., Jing, L., Bin, Y. & Zhongzhen, Y. (2010), An adaptive bus arrival time prediction model, *Journal of the Eastern Asia Society for Transportation Studies*, **8**, 1126–36.

Breiman, L. (2001), Random forests, *Machine Learning*, **45**(1), 5–32.

Breiman, L. I., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984), Classification and regression trees (CART), *Lecture Notes in Computer Science*, **40**(3), 17–23.

Chan, K., Lam, W. & Tam, M. (2009), Real-time estimation of arterial travel times with spatial travel time covariance relationships, *Transportation Research Record: Journal of the Transportation Research Board*, **2121**, 102–109.

Chang, H., Park, D., Lee, S., Lee, H. & Baek, S. (2010), Dynamic multi-interval bus travel time prediction using bus transit data, *Transportmetrica*, **6**(1), 19–38.

Chen, M., Liu, X., Xia, J. & Chien, S. I. (2004), A dynamic bus-arrival time prediction model based on APC data, *Computer-Aided Civil and Infrastructure Engineering*, **19**(5), 364–76.

Chien, S. I.-J., Ding, Y. & Wei, C. (2002), Dynamic bus arrival time prediction with artificial neural networks, *Journal of Transportation Engineering*, **128**(5), 429–38.

Chien, S. I.-J. & Kuchipudi, C. M. (2003), Dynamic travel time prediction with real-time and historic data, *Journal of Transportation Engineering*, **129**(6), 608–16.

Cristianini, N. & Shawe-Taylor, J. (2000), *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, United Kingdom.

Cutler, D. R., Edwards Jr., T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J. & Lawler, J. J. (2007), Random forests for classification in ecology, *Ecology*, **88**(11), 2783–92.

Dharia, A. & Adeli, H. (2003), Neural network model for rapid forecasting of freeway link travel time, *Engineering Applications of Artificial Intelligence*, **16**(7), 607–13.

Dietterich, T. G. (2002), Ensemble learning, *The Handbook of Brain Theory and Neural Networks*, **2**, 110–25.

Friedman, J. H. & Meulman, J. J. (2003), Multiple additive regression trees with application in epidemiology, *Statistics in Medicine*, **22**(9), 1365–81.

Gal, A., Mandelbaum, A., Schnitzler, F., Senderovich, A. & Weidlich, M. (2015), Traveling time prediction in scheduled transportation with journey segments, *Information Systems*, **64**(C), 266–80.

Genuer, R., Poggi, J.-M. & Tuleau-Malot, C. (2010), Variable selection using random forests, *Pattern Recognition Letters*, **31**(14), 2225–36.

Ghimire, B., Rogan, J. & Miller, J. (2010), Contextual land-cover classification: incorporating spatial dependence in land-cover classification models using random forests and the Getis statistic, *Remote Sensing Letters*, **1**(1), 45–54.

Gislason, P. O., Benediktsson, J. A. & Sveinsson, J. R. (2006), Random forests for land cover classification, *Pattern Recognition Letters*, **27**(4), 294–300.

Grimm, R., Behrens, T., Märker, M. & Elsenbeer, H. (2008), Soil organic carbon concentrations and stocks on Barro Colorado Island—digital soil mapping using Random Forests analysis, *Geoderma*, **146**(1), 102–13.

Hamner, B. (2010), Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow, in *Paper presented at the 2010 IEEE International Conference on Data Mining Workshops (ICDMW)*, Sydney, NSW, Australia.

Ho, T. K. (2002), A data complexity analysis of comparative advantages of decision forest constructors, *Pattern Analysis & Applications*, **5**(2), 102–12.

Iverson, L. R., Prasad, A. M., Matthews, S. N. & Peters, M. (2008), Estimating potential habitat for 134 eastern US tree species under six climate scenarios, *Forest Ecology and Management*, **254**(3), 390–406.

Jeong, R. & Rilett, L. R. (2004), Bus arrival time prediction using artificial neural network model, in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, Washington DC, 988–93.

Jeong, R. H. (2005), The prediction of bus arrival time using automatic vehicle location systems data, Texas A&M University, College Station, TX.

Jiang, X. & Adeli, H. (2004), Clustering-neural network models for freeway work zone capacity estimation, *International Journal of Neural Systems*, **14**(03), 147–63.

Jiang, X. & Adeli, H. (2005), Dynamic wavelet neural network model for traffic flow forecasting, *Journal of Transportation Engineering*, **131**(10), 771–79.

Kalman, R. E. (1960), A new approach to linear filtering and prediction problems, *Journal of Basic Engineering*, **82**(1), 35–45.

Leshem, G. & Ritov, Y. (2007), Traffic flow prediction using Adaboost algorithm with random forests as a weak learner, *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, **1**(1), 1–6.

Liaw, A. & Wiener, M. (2002), Classification and regression by Random Forest, *R News*, **2**(3), 18–22.

Ließ, M., Glaser, B. & Huwe, B. (2012), Uncertainty in the spatial prediction of soil texture: comparison of regression tree and random forest models, *Geoderma*, **170**, 70–79.

McLachlan, G., Do, K.-A. & Ambroise, C. (2005), *Analyzing Microarray Gene Expression Data*, vol. 422, John Wiley & Sons, Hoboken, NJ.

Moreira, J. P. C. L. M. (2008), Travel time prediction for the planning of mass transit companies: a machine learning approach, Universidade do Porto, Porto, Portuguese Republic.

Park, D. C., El-Sharkawi, M., Marks, R., Atlas, L. & Damborg, M. (1991), Electric load forecasting using an artificial neural network, *IEEE Transactions on Power Systems*, **6**(2), 442–49.

Park, S. H., Jeong, Y. J. & Kim, T. J. (2007), Transit travel time forecasts for location-based queries, *Journal of the Eastern Asia Society for Transportation Studies*, **7**, 1859–69.

Patnaik, J., Chien, S. & Bladikas, A. (2004), Estimation of bus arrival times using APC data, *Journal of Public Transportation*, **7**(1), 1–20.

Peters, J., Verhoest, N., Samson, R., Boeckx, P. & De Baets, B. (2008), Wetland vegetation distribution modelling for the identification of constraining environmental variables, *Landscape Ecology*, **23**(9), 1049–65.

Sesnie, S. E., Gessler, P. E., Finegan, B. & Thessler, S. (2008), Integrating Landsat TM and SRTM-DEM derived variables with decision trees for habitat classification and change detection in complex neotropical environments, *Remote Sensing of Environment*, **112**(5), 2145–59.

Shalaby, A. & Farhan, A. (2003), Bus travel time prediction model for dynamic operations control and passenger information systems, in *Paper prepared for presentation at the 82nd Annual Meeting of the Transportation Research Board*, Washington DC, January 2003.

Smith, B. L., Williams, B. M. & Oswald, R. K. (2002), Comparison of parametric and nonparametric models for traffic flow forecasting, *Transportation Research Part C: Emerging Technologies*, **10**(4), 303–21.

Steele, B. M. (2000), Combining multiple classifiers: an application using spatial and remotely sensed information for land cover type mapping, *Remote Sensing of Environment*, **74**(3), 545–56.

Tam, M. L. & Lam, W. H. (2009), Short-term travel time prediction for congested urban road networks, in *Paper presented at the Transportation Research Board 88th Annual Meeting*, Washington DC.

Thomas, T., Weijermars, W. & Van Berkum, E. (2010), Predictions of urban volumes in single time series, *IEEE Transactions on Intelligent Transportation Systems*, **11**(1), 71–80.

van Hinsbergen, C. I., Van Lint, J. & Van Zuylen, H. (2009), Bayesian committee of neural networks to predict travel times with confidence intervals, *Transportation Research Part C: Emerging Technologies*, **17**(5), 498–509.

Vapnik, V. (2013), *The Nature of Statistical Learning Theory Neural Networks*. Springer Science & Business Media, New York, NY.

Vapnik, V. N. (1999), An overview of statistical learning theory, *IEEE Transactions on Neural Networks*, **10**(5), 988–99.

Wall, Z. & Dailey, D. (1999), An algorithm for predicting the arrival time of mass transit vehicles using automatic vehicle location data, in *Paper presented at the 78th Annual Meeting of the Transportation Research Board, National Research Council*, Washington DC.

Williams, B. M. & Hoel, L. A. (2003), Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results, *Journal of Transportation Engineering*, **129**(6), 664–72.

Wu, M. & Adeli, H. (2001), Wavelet-neural network model for automatic traffic incident detection, *Mathematical and Computational Applications*, **6**(2), 85–96.

Yang, R.-M., Zhang, G.-L., Liu, F., Lu, Y.-Y., Yang, F., Yang, F., Yang, M., Zhao, Y. G. & Li, D.-C. (2016), Comparison of boosted regression tree and random forest models for mapping topsoil organic carbon concentration in an alpine ecosystem, *Ecological Indicators*, **60**, 870–78.

Yu, B., Yang, Z.-Z., Chen, K. & Yu, B. (2010), Hybrid model for prediction of bus arrival times at next station, *Journal of Advanced Transportation*, **44**(3), 193–204.

Yu, B., Yang, Z. & Yao, B. (2006), Bus arrival time prediction using support vector machines, *Journal of Intelligent Transportation Systems*, **10**(4), 151–58.

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Computer-Aided Civil and Infrastructure Engineering **32** (2017) 154–169

Short-Term Traffic Speed Prediction for an Urban Corridor

Baozhen Yao, Chao Chen, Qingda Cao, Lu Jin & Mingheng Zhang

School of Automotive Engineering, Dalian University of Technology, Dalian 116024, P. R. China

Hanbing Zhu & Bin Yu*

School of Transportation Science and Engineering, Beihang University, Beijing 100191, P. R. China and Transportation Management College, Dalian Maritime University, Dalian 116026, P. R. China

Abstract: *Short-term traffic speed prediction is one of the most critical components of an intelligent transportation system (ITS). The accurate and real-time prediction of traffic speeds can support travellers' route choices and traffic guidance/control. In this article, a support vector machine model (single-step prediction model) composed of spatial and temporal parameters is proposed. Furthermore, a short-term traffic speed prediction model is developed based on the single-step prediction model. To test the accuracy of the proposed short-term traffic speed prediction model, its application is illustrated using GPS data from taxis in Foshan city, China. The results indicate that the error of the short-term traffic speed prediction varies from 3.31% to 15.35%. The support vector machine model with spatial-temporal parameters exhibits good performance compared with an artificial neural network, a k-nearest neighbor model, a historical data-based model, and a moving average data-based model.*

1 INTRODUCTION

The urbanization and mobilization occurring in China is causing severe traffic congestion in most metropolises. There is no way to completely satisfy the demand by constructing new lanes or roads because of limited urban land resources. Therefore, it is important to improve the efficiency of the existing road network. In recent years, intelligent transportation system (ITS) technologies have been widely applied in China, which are essential components of a traffic management system. Monitoring facilities and positioning equipment (for instance, GPS equipped on taxis and private cars) have been installed in most cities. To implement advanced and efficient traffic management systems, the increasing need for short-term traffic speed prediction has attracted the attention of traffic engineers and researchers.

Many relevant techniques of intelligent transportation systems have successfully been applied in the highway management and traffic delay optimization models, such as neural networks, wavelets, and chaos theory (Ghosh-Dastidar and Adeli, 2003, 2006; Karim and Adeli, 2003a, b; Jiang and Adeli, 2003, 2004a, b). An extensive review of short-term traffic prediction was provided by Vlahogianni et al. (2004). The two well-known modelling approaches for short-term prediction methods can broadly be classified into parametric and nonparametric techniques (Chen et al., 2012, 2013; Chen and Yang, 2010). Among the wide variety of available statistical parametric techniques, several algorithms have been applied in short-term traffic flow prediction, including a historical average algorithm (Smith and Demetsky, 1997), smoothing techniques (Smith and Demetsky, 1997; Williams et al., 1998) and the autoregressive integrated moving average (ARIMA) (Kirby et al., 1997; Hamed et al., 1995; Williams, 2001). A radial basis function neural network (RBFNN) learns input–output mapping by covering the input space with basis functions that transform a vector from the input space to the output space (Adeli and Karim, 2000; Adeli, 2001; Karim and Adeli, 2002, 2003c; Adeli and Jiang, 2003; Dharia and Adeli, 2003). Statistical techniques were popular in short-term traffic flow prediction in the 1990s because

*To whom correspondence should be addressed. E-mail: *ybzhyb@163.com*.

DOI: 10.1111/mice.12221

of their simple structure and rapid, low-cost updating. Wavelet-based signal processing is a powerful tool for the analysis and synthesis of time series (Mallat, 1999). The wavelet packet-autocorrelation function (ACF) was proposed for the selection of the decomposition level in the wavelet multiresolution analysis of traffic flow time series (Jiang, 2004b). Kalman filtering theory, which uses a state-space model, has also been applied in short-term traffic flow prediction because of its use by Okutani and Stephanedes (1984). In previous studies, time series-based methods have been proposed for predicting traffic flow (Wu et al., 2004; Kayacan et al., 2010). The results from multiple studies (Okutani and Stephanedes, 1984; Smith and Demetsky, 1996; Stathopoulos and Karlaftis, 2003; Ojeda et al., 2013) demonstrated that Kalman filter theory is well-suited for modelling transportation data because of its multivariate nature. Moreover, the Kalman filter algorithm allows the state variable to be updated continuously. This is the main reason that the Kalman filter algorithm is a popular short-term traffic flow prediction method. In addition, Kalman filtering theory has been proposed for predicting short-term traffic conditions (Wang and Papageorgiou, 2005; Antoniou et al., 2007). Nagel et al. (2000) contributed some original and important high-speed microsimulations of large-scale road networks.

Nonparametric regression relies on the relationship between dependent and independent variables rather than on any specific functional form. Nonparametric techniques, such as support vector machines (SVMs), artificial neural networks (ANNs), and the *k*-nearest neighbor (*k*-NN) model, have attracted attention for short-term traffic prediction. Applications of neural networks to short-term traffic prediction are based on simple multilayer perceptrons (MLPs) (Dharia and Adeli, 2003; Smith and Demetsky, 1995, 1997; Clark, 2003; Dia, 2001; Dougherty and Cobett, 1997; Ishak et al., 2003). Some researchers have introduced other techniques to improve the prediction performance of the neural network algorithm, e.g., the wavelet microsimulation model (Ghosh-Dastidar, 2006); Boltzmann-simulated annealing (Jiang, 2003); Kalman filter (Vythoulkas, 1993); wavelet (Boto-Giralda et al., 2010; Jiang and Adeli, 2005); moving average model, exponential smoothing model and autoregressive MA (ARIMA) (Tan et al., 2009); Bayesian model (Zheng et al., 2006); and empirical mode decomposition (Wei and Chen, 2012). The *k*-NN model is one of the most popular methods because of its simple nature and wide applicability, and it has been successfully applied by many scholars for short-term traffic flow prediction (Smith et al., 2002; Zuo et al., 2008; Akbari et al., 2011; Turochy, 2006; Lam et al., 2006; Chang et al., 2012; Zhang et al., 2013). These studies demonstrate that the *k*-NN model performs well in predicting short-term traffic flow. Recently, new interest in SVMs (Zhang and Xie, 2008; Wu et al., 2004; Manoel et al., 2009; Zhang and Wu, 2009, 2012a, 2012b) has arisen for short-term traffic prediction. A SVM is a mathematical data-driven model similar to the ANN model that can identify a complicated nonlinear system and does not require a formula derived from existing data. Wu et al. (2004) and Yu et al. (2006) applied support vector regression (SVR) for travel-time prediction. By comparing its results to those of other baseline travel-time prediction methods, they examined the feasibility and applicability of SVMs for vehicle travel-time prediction.

Researchers have proposed many methods for short-term traffic speed prediction. However, a traveller would prefer to be given the likely traffic speed of the whole route (including multiple road links) from origin to destination at the time when he/she will be on the roads rather than further traffic information on a road link in the urban corridor. If the traffic speed of the route between the origin and the destination could be predicted, travellers would be able to efficiently select an urban corridor as soon as they are on the road, and administrators would be able to successfully manage, control, and guide traffic speed on the road network. To provide further traffic information on the urban corridor (including multiple road links) to travellers, scholars have applied historical and real-time traffic information-based models (Yildirimoglu and Geroliminis, 2013), an extended time-series-based approach (Min and Wynter, 2011), modular neural networks (Vlahogianni et al., 2007), and dynamic Bayesian networks (Queen and Albers, 2009) to predict traffic speed at multiple road link locations. Additionally, Hofleitner et al. (2012) used a hybrid model to predict traffic speed.

However, for traffic speed prediction along an entire corridor, it is not sufficient to sum the predicted travel times (or instantaneous travel times) of road links included in the corridor at the starting time of the trip. Because cars spend time going through each road link, when a car arrives at the next road link, the traffic speeds will not be the same as those predicted before. Hence, there are two steps to predict the travel time of a road link correctly: (1) predicting the arrival time at the target road link and (2) predicting the driving time of the target road link at the time when the car will arrive there.

This article aims to make two contributions. First, it focuses on using SVMs in short-term traffic speed prediction. This prediction model can support travellers' route choices and traffic guidance. A single-step prediction model considering spatial and temporal parameters is proposed. Following this, a short-term traffic

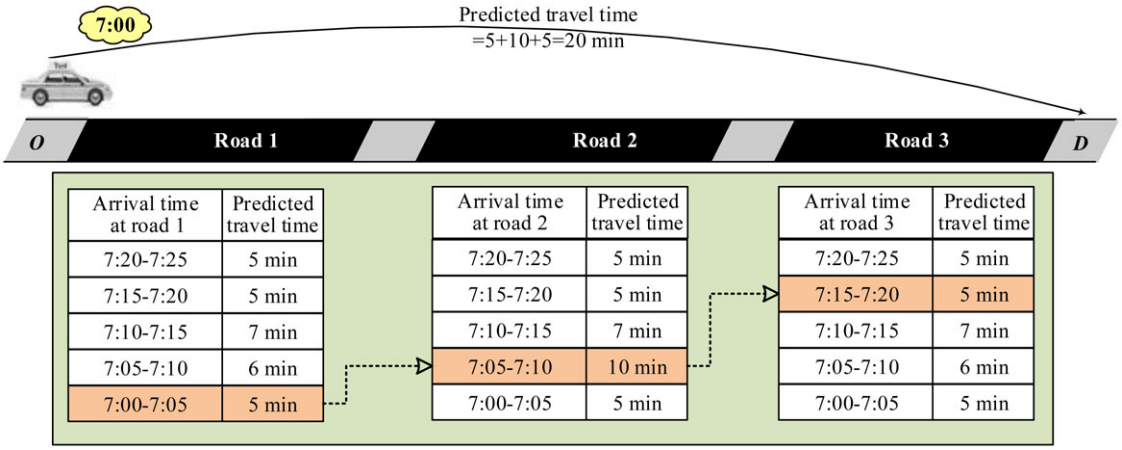


Fig. 1. Short-term traffic speed prediction.

speed prediction model, including the multi-time-step traffic prediction of several road links, is developed. Second, the performance of several prediction methods (namely, SVM, ANN, *k*-NN, ARIMA, the historical data-based model and the moving average data-based model) are assessed and compared in terms of prediction traffic speed. The performance comparison of several models can provide valuable insight for researchers as well as practitioners.

The remainder of the article is organized as follows. Section 2 describes the short-term traffic speed prediction problem. In Section 3, the theory underlying the SVM algorithm is introduced. A single-step prediction model of short-term traffic speeds, in which spatial-temporal parameters are considered, is proposed in Section 4. Then, the short-term traffic speed prediction model is constructed. Data collection and processing are described in Section 5. Section 6 reports the computational results, and the conclusions are discussed in Section 7. Finally, a list of symbols is given in Section 8.

2 PROBLEM DESCRIPTION

When people want to reach their destination quickly, they tend to choose the fastest route. However, because of changing traffic speed, decisions cannot be made merely according to the spatial distance from the origin to the destination. Therefore, time-dependent traffic speed information between the origin and the destination is particularly important for travellers making flexible route choices.

Short-term traffic speed prediction can predict traffic speeds for more than one road link. For instance, a car plans to travel from an origin to a destination (Figure 1), and the driver cares about the length of the trip. The arrival time of the car at each road link is not fixed because of the time-varying characteristic of traffic speed. Thus, at different arrival times at a road link, the traffic speeds are different.

Figure 1 shows an example of a short-term traffic speed prediction. The car arrives at the beginning of road link 1 at 7:00, and the driver wants to know how long it will take from the beginning of road link 1 through three road links to the destination. First, a multi-time-step prediction of each road link can be implemented through a single-step prediction model, and the predicted travel time (i.e., the estimated travel time of each period between 7:00 and 7:25) can be obtained. Assume that the travel time at road link 1 of the car is 5 min. Then, the car will reach road link 2 at 7:05. The travel time of the car at road link 2 is estimated to be approximately 10 min between 7:05 and 7:10. Thus, the car will reach road link 3 at 7:15 and the destination at 7:20.

3 SVM MODEL

The SVM was originally used in pattern recognition. With the introduction of the ε-insensitive loss function by Vapnik, the SVM has been extended to solve the problem of nonlinear regression. The support vector regression machine first needs to choose a nonlinear mapping $\Phi(x)$, which can map data in the original space to a high-dimensional feature space and then make a linear estimate in the high-dimensional feature space. Assume a set of training samples $\{x_i, y_i\}_i^n$, for which the input data $x_i \in R^N$ and output data $y_i \in R$, and construct the optimal decision function in high-dimensional feature space:

$$f(x) = w^T \Phi(x) + b \tag{1}$$

where Φ (x) is a nonlinear mapping that can convert all raw data to another feature space in which the sample data are linearly separable, w is the weight and b is the offset, which can be estimated by minimizing Equation (2):

$$\min R_{SVM}(C) = \frac{C}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) + \frac{1}{2}\|w\|^2 \quad (2)$$

Define the ε-insensitive loss function:

$$L(y, f(x)) = \begin{cases} 0 & \text{if } |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon & \text{otherwise} \end{cases} \quad (3)$$

$L(y, f(x))$ is the loss function, which is used to measure the degree of prediction error. For a given input x_i, $f(x)$ gives the corresponding result, which may be different from y. The ε-insensitivity can allow the presence of prediction error within a certain range and ensure that the model can find the optimal solution. Empirical risk is the difference between predicted and real value. The first part of the empirical risk on the right in Equation (2) can be estimated by the nonsensitive loss function given in Equation (3), and the second part is the regularization confidence. Regularization confidence is a protection method that can avoid overfitting. Regularization confidence works by introducing certain restrictions that can reduce the complexity of the machine learning model. The structural risk minimization principle of the SVM is a compromise considering the empirical risk and confidence limit, minimizing the expected risk and preventing overlearning problems. The value of ε affects the support vector size, and C is the regularization parameter, which controls the degree of punishment beyond the error of the sample. By introducing the relaxation of nonnegative variables ξ_i and ξ_i^*, the objective function equation of the support vector regression machine (2) can be transferred into Equation (4):

$$\min R(w, b) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$
$$s.t. \begin{cases} y_i - w^T\Phi(x_i) - b \le \varepsilon + \xi_i^* \\ w^T\Phi(x_i) + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i \ge 0, \xi_i^* \ge 0 \end{cases} \quad (4)$$

Finally, by introducing the Lagrange multiplier, the optimization problem is converted into a dual problem:

$$R(a_i, a_i^*) = \sum_{i=1}^{n} d_i(a_i - a_i^*) - \varepsilon\sum_{i=1}^{n}(a_i + a_i^*) - \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(a_i - a_i^*)(a_j - a_j^*)K(x_i, x_j) \quad (5)$$

$$s.t. \begin{cases} \sum_{i=1}^{n}(a_i - a_i^n) = 0 \\ 0 \le a_i, a_i^* \le C, i = 1, \ldots n \end{cases}$$

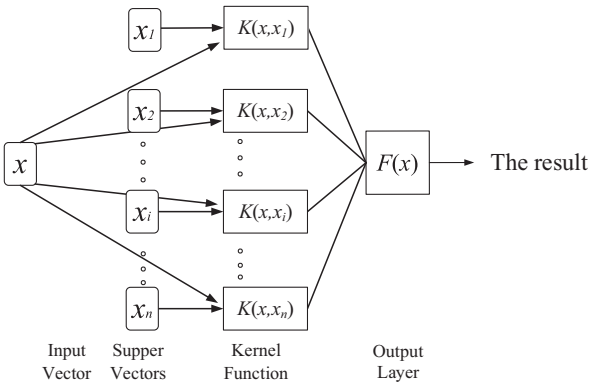


Fig. 2. Structure of SVM.

The decision function shown in Equation (1) is correspondingly converted into:

$$R(a_i, a_i^*) = \sum_{i=1}^{n}(a_i - a_i^*)K(x, x_i) + b \quad (6)$$

In Equation (6), $K(x, x_i)$ is the kernel function. Essentially, the kernel function is a mapping function. To reduce the algorithm complexity by using the kernel function, kernel functions can convert a nonlinear learning problem into a linear learning problem. The parameters a_i, a_i^* are the corresponding Lagrange multipliers, where $\alpha_i a_i^* = 0$ only if the corresponding data sample point of $a_i \ne a_i^*$ is defined as the SVM. SVM shows a strong resistance to the overfitting problem and offers excellent generalization performance. This is mainly because SVM can construct a mapping from one-dimensional input vector into high-dimensional space by the use of reproducing kernels. The architecture of SVM is shown in Figure 2.

4 SHORT-TERM TRAFFIC SPEED PREDICTION MODEL

In an urban road network, road links do not exist in isolation. Traffic speed on both upstream and downstream road links can affect the traffic speed of the current road link. An understanding of both spatial and temporal information facilitates identifying road information.

Based on the successful cases mentioned above, an improved SVM model composed of temporal and spatial parameters is developed in our study. First, single-step prediction models including different state vectors are established, and then a short-term traffic speed prediction model is developed.

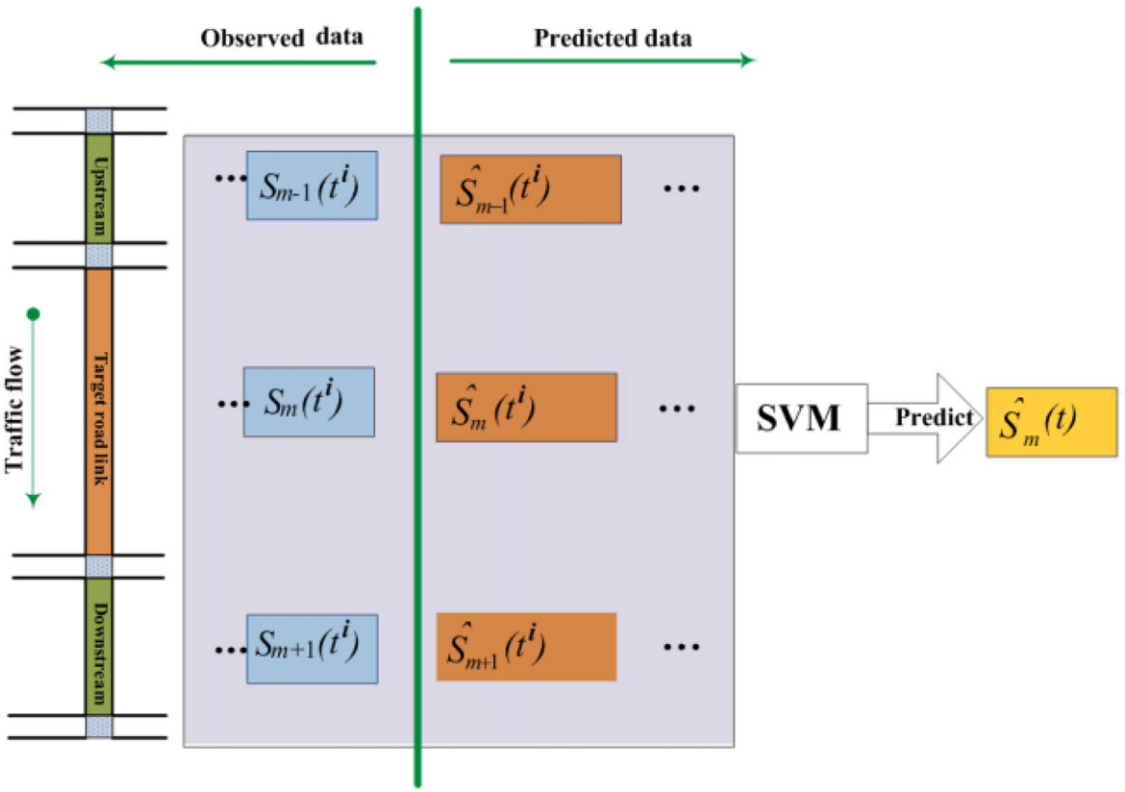


Fig. 3. Prediction model with spatial and temporal information based on SVM.

4.1 Single-step prediction model based on SVM

In the single-step model, the temporal information of the target road link and traffic speed of upstream/ downstream road links is considered. Figure 3 shows the structure of the spatial-temporal model.

Assume the target road link is the m road link and that the current time is the t interval. In the article, the speed at each road link is used to estimate the traffic speed $\hat{S}_m(t)$. The parameter Φ^{T} denotes the temporal state vector of the target road link, Φ^{U} denotes the traffic speed of the upstream road links, and Φ^{D} denotes the traffic speed of the downstream road links.

State vectors: $\mathbf{X} = \{\Phi^{\mathrm{T}}, \Phi^{\mathrm{U}}, \Phi^{\mathrm{D}}\}$

$$\boldsymbol{\Phi}^T = \{\cdots, S_{m-1}(t^i),\ \hat{S}_{m-1}(t^i), \cdots\}$$

$$\boldsymbol{\Phi}^T = \{\cdots, S_m(t^i),\ \hat{S}_m(t^i), \cdots\}$$

$$\boldsymbol{\Phi}^T = \cdots, S_{m+1}(t^i), \hat{S}_{m+1}(t^i), \cdots$$

Output values: $\hat{S}_m(t)$

where m is the mth target road link, t is the time period of vehicle entering within mth road link, t^i is the ith time period; $S_m(t)$ is the actual speed on mth road link in time period t; $\hat{S}_m(t)$ is the prediction speed of vehicle on mth road link in time period t.

4.2 Short-term traffic speed prediction model

A short-term traffic speed prediction model predicts the traffic speed on a route containing more than one road link. In a short-term traffic speed prediction model, the traffic speed of each road link included in a route will be first predicted based on a single-step prediction model. Then, the car's arrival time at the 2nd road link is predicted, and the traffic speeds at the 2nd road link at the car's arrival time can be estimated. This process is repeated to compute the destination arrival time. In addition, if the travel time changes from time period t to time period t+1 during travel, the $\hat{S}_m(t)$ in time period t is used rather than t+1. The reason is that the speed of the target car is primarily influenced by the traffic condition in front of the target car rather than the traffic condition behind the target car. Figure 4 shows the procedure of the short-term traffic speed prediction model. Assume that the route includes M road links. The mathematical expression of the short-term traffic speed prediction model is as follows:

$$\Gamma = T_1 + \hat{T}_1 + \hat{T}_2 + \cdots + \hat{T}_M = T_1 + \sum_{m=1}^{M} \hat{T}_m \qquad (7)$$

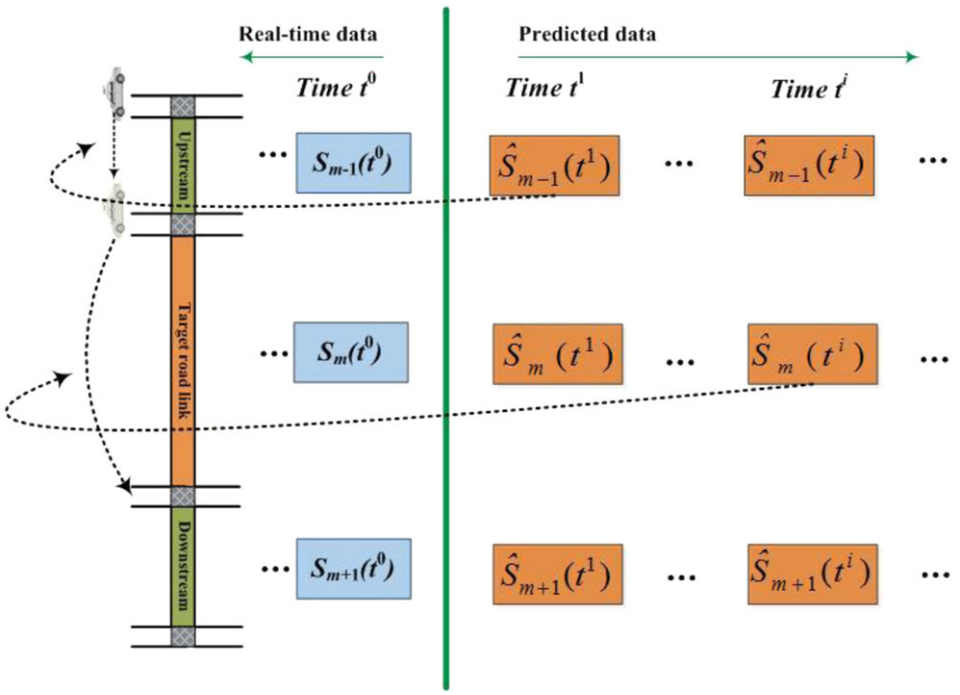


Fig. 4. Short-term traffic speed prediction model.

$$\hat{T}_m = \frac{L_m}{\hat{S}_m(t)} \tag{8}$$

$$t = \frac{T_m - T_1}{t_d} \tag{9}$$

$$T_m = T_1 + \hat{T}_1 + \cdots + \hat{T}_{m-1} \tag{10}$$

where Γ is the travel time from beginning to end; M is the number of the links between the origin and the destination; m is the mth road link; t_d is the time interval; T_1 is the time of vehicle arrival at 1st road link; T_m is the time of vehicle arrival at mth road link; $\hat{T}_m$ is the travel time of vehicle entering within mth road link; L_m is the length of mth road link; t is the time period of vehicle entering within mth road link; $\hat{S}_m(t)$ is the prediction speed of vehicle in time period t on mth road link; "[·]" is used to round up to an integer.

The calculation steps are as follows:

***Step* 1** Initialization:

First, data on the current road link, upstream road link and downstream road link are gathered.
Set $m = 1$, where m denotes the current road link.
Set M to be the maximum number of links between the origin and the destination.

***Step* 2** Establish a single-step prediction model for each road link.

***Step* 3** Predict the travel time of the road link. (The arrival time at the entry to the next road link can also be determined.)

***Step* 3.1** Prepare the input data according to the arrival time of the current road link; the input data may be actual data from a historical database or data predicted previously.

***Step* 3.2** Use the single-step prediction model to predict the travel time $\frac{L_m}{\hat{S}_m(t)}$ of the current road link.

***Step* 3.3** Combining the entry time m and driving time $\frac{L_m}{\hat{S}_m(t)}$ of the current road link, calculate the arrival time $\hat{T}_{m+1}$ at the entry to the next road link m+1.

***Step* 4** Termination check of the multi-time-step prediction.

If exceeding the maximum number ($m > M$), then stop. Otherwise, set $m = m$+1 and move on to *Step* 3.

5 DATA COLLECTION AND PROCESSING

Collecting real-time traffic data is essential for short-term traffic speed prediction. The wide application of GPS equipment in taxis represents an easy method to obtain first-hand data of floating cars. Large quantities of real-time data can be collected from the GPS system, such as the longitude and latitude of a running car, travel speed, driving direction, and time. In this article,

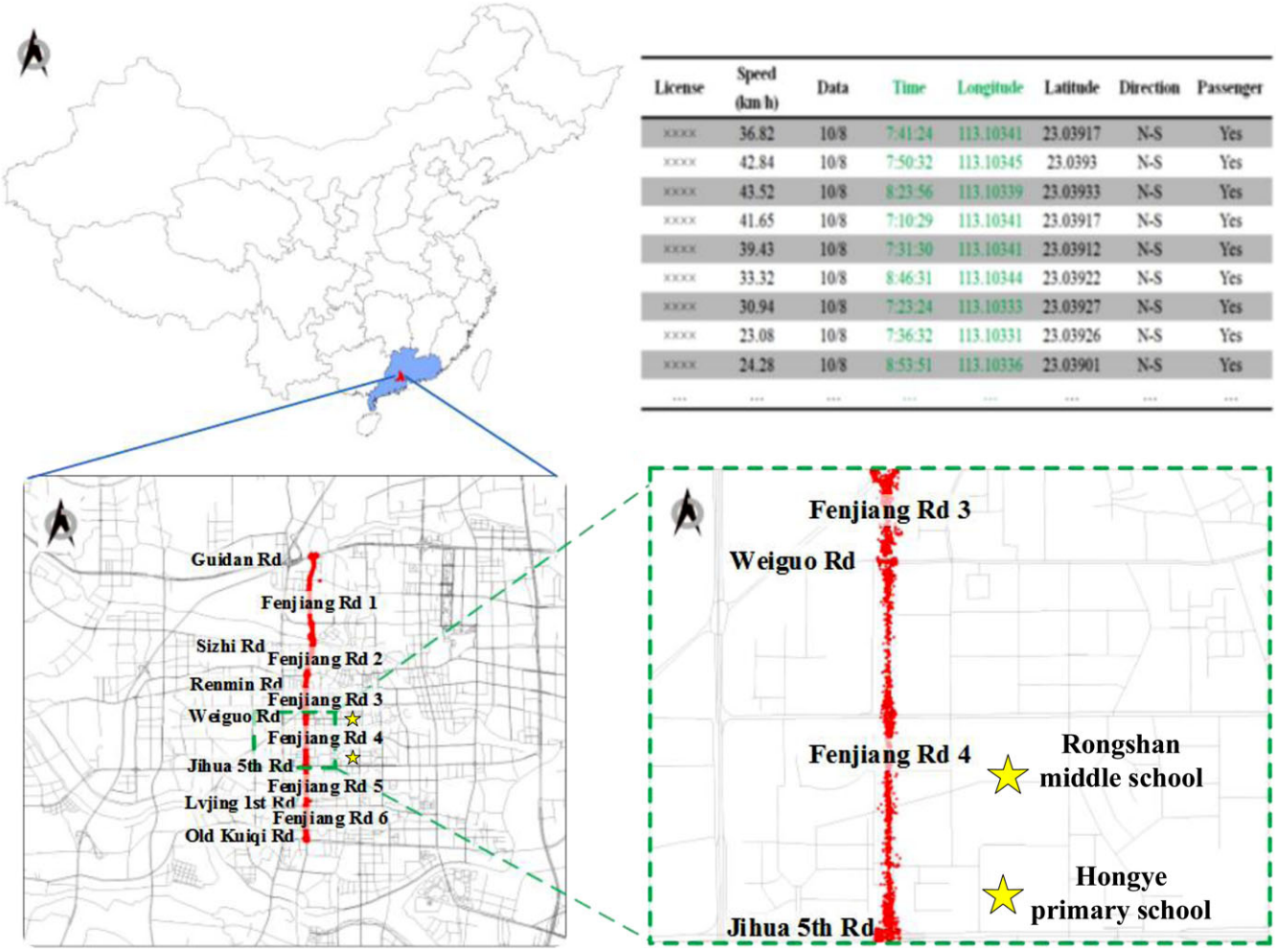


Fig. 5. Spatial locations of the research targets and the information obtained from GPS.

Table 1
Information of Fenjiang road

Road name	*Road interval*	*Detail*	*Length (Km)*
Fenjiang Rd 1	Guidan Rd-Sizhi Rd	4 lane, two-way	2.85
Fenjiang Rd 2	Sizhi Rd-Renmin Rd	4 lane, two-way	0.84
Fenjiang Rd 3	Renmin Rd-Weiguo Rd	4 lane, two-way	0.79
Fenjiang Rd 4	Weiguo Rd-Jihua 5th Rd	4 lane, two-way	1.29
Fenjiang Rd 5	Jihua 5th Rd-Lvjing 1st Rd	4 lane, two-way	1.07
Fenjiang Rd 6	Lvjing 1st Rd-Old Kuiqi Rd	4 lane, two-way	0.87

we compute the traffic speeds based on taxi GPS data in Foshan city, China.

Six road links on Fenjiang Road in Foshan city are chosen as the study objects: Guidan Road–Sizhi Road, Sizhi Road–Renmin Road, Renmin Road—Weiguo Road, Weiguo Road–Jihua 5th Road, Jihua 5th Road–Lvjing 1st Road, and Lvjing 1st Road–Old Kuiqi Road. The spatial location and road information of the six road links are shown in Figure 5 and Table 1.

The GPS systems of the sampling taxis can send information every 30 s. The probability that the information cannot be transmitted is less than 10%. The information includes the longitude, latitude, travel speed, driving direction, and time. During workdays between

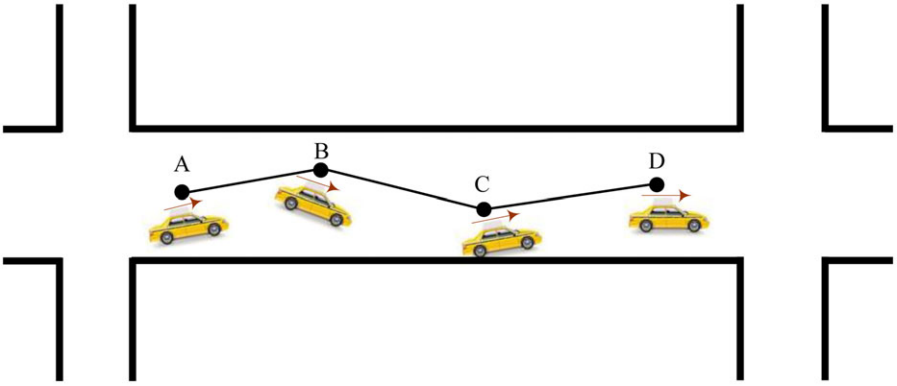


Fig. 6. An example of the speed computation of a taxi.

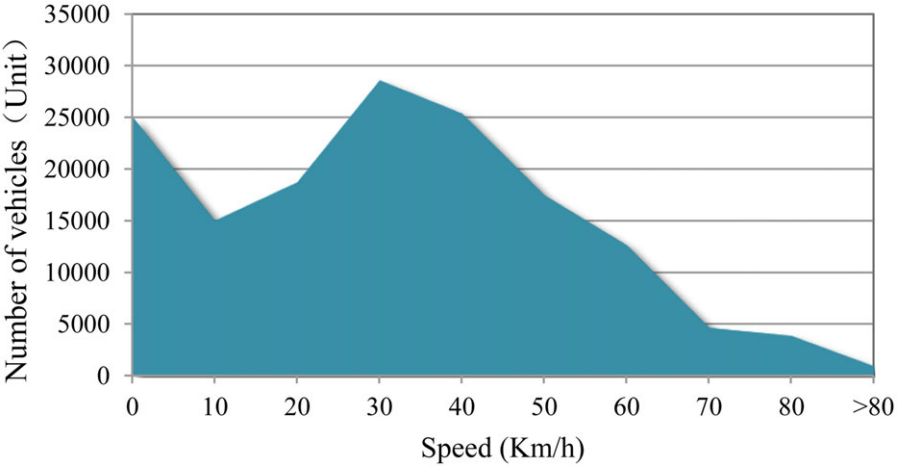


Fig. 7. Distribution of driving speed.

October 8 and November 10 in 2012, GPS information from 400 taxis was collected during the morning peak (7:00–9:00), and we selected six road links with sufficient data. Figure 5 shows the locations of the taxis by matching the longitude and latitude information to the map.

Although speed data for the taxis are included in the GPS information, the speed values are unreliable because the speed is a vector with direction. Thus, after map-matching, the speed of the taxi must be recalculated according to the location information of the taxi. Figure 6 shows a simple example of the speed computation of a taxi using location information and time data. Points A, B, C, and D are the positions at which the taxi sends GPS data during the course of its movement on a road link. Then, the speed of the taxi from points A to D can be calculated using the distance and time gaps between these points. Note that the distance from points A to D is obtained by summing the spherical distances between each set of two neighboring points rather than by the Euclidean distance between points A and D. The spherical distance is calculated using the data from the GPS system. Moreover, Equation (11) is used to calculate the spherical distance between two adjacent points. The GPS sends data every 30 s, and as a result, the data are not consecutive. The arrival time and departure time of the road links are not known. In Figure 6, point A is not the beginning of the road link, and point D is not the terminal point. However, these are the first and last points at which the car transmitted data. We use the distance between A and D to divide the time interval between them to obtain the travel speed of the road link.

$$d = r * arccos\left\{\sin(x_1) * \sin(x_2) + \cos(x_1) * \cos(x_2) * \cos(y_1 - y_2)\right\} \quad (11)$$

where d is the spherical distance, r is the radius of the earth, x_1 is the dimension of A, y_1 is the precision of A, x_2 is the dimension of D, and y_2 is the precision of D.

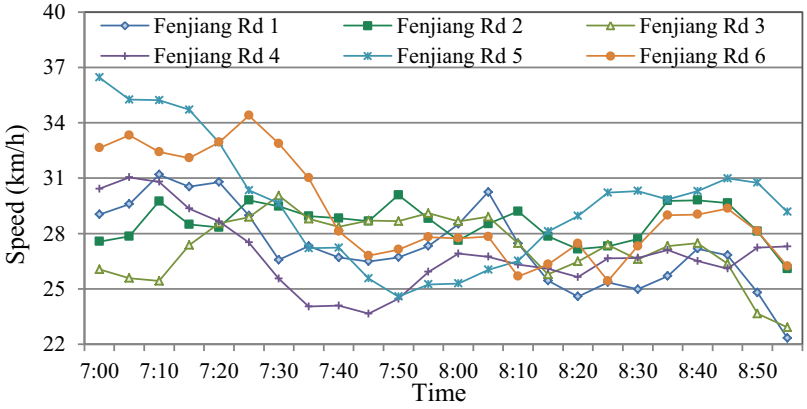


Fig. 8. Average speed on six road links during different periods.

However, data from the GPS are not always accurate. (The accuracy is approximately 90%.) Data distortion may occur either because of detector malfunction or transmission problems, and thus, abnormal data should be eliminated by calculating the driving speed. All driving speeds exceeding 60 km/h are eliminated because the speed limit of the target road link is 60 km/h. Moreover, the speed of an empty taxi cannot represent the reality of traffic speed on the target road link. Therefore, the data on an empty taxi were not used in this study. When the taxi stops to discharge a passenger, the speed is zero; such data are discarded, and data use resumes only when the taxi moves again. Finally, Equation (12) is used to calculate the space-mean speed on each road link.

$$S_m(t) = \sum_{i=1}^{N} V_i(t)/N \tag{12}$$

where $S_m(t)$ is the average driving speed at time t on road link m, N is the number of times that taxi sends GPS data on the road link m, and $V_i(t)$ is the speed of the taxi at each position. Thus, the average driving speed on a road link during each time period can be obtained. In total, 118,067 valid data points were obtained. The distribution of driving speed is shown in Figure 7. Figure 8 shows the average speed on each road link over different time periods. As seen Figure 8, there is a large fluctuation in the observed speed on Fenjiang Rd 4. The primary reason is that there are two schools, Rongshan middle school and Hongye primary school (Figure 5), along Fenjiang Rd 4. Because there is no school bus for the two schools, parents need to drop their children off at school. Available parking spots are randomly distributed, which leads to increased variation in the observed speeds on the road.

6 NUMERICAL STUDY

The database is divided into three categories: sample set, test set, and prediction set. This study was conducted on a PC with an Inter Core i5 CPU (2.5 GHz) and 4 GB memory. For the one month of taxi data in Foshan, those from Monday to Wednesday are taken as the sample data (past data set for calibrating parameters and validating the prediction performance), whereas those from Thursday and Friday are taken as the test data (used to calibrate the parameters of the models) and those from Saturday and Sunday are taken as prediction data (used to validate the prediction performance of the models). Overall, the test data and prediction data constitute approximately 20% of the sample database, and the rest are used as a sample set. The mean absolute percentage error (MAPE) was used as a measure to evaluate the efficacy of the proposed models in this article.

$$MAPE = \frac{1}{P}\sum_{i=1}^{P}\frac{\left|\hat{S}_m(t) - S_m(t)\right|}{S_m(t)} \tag{13}$$

where $\hat{S}_m(t)$ is the prediction speed of vehicle on road link m in time period t, $S_m(t)$ is the actual speed on road link m in time period t, and P is the prediction number.

6.1 Parameter determination

To establish the single-step model, we validate the performances of various models with different state vectors, as shown in Table 2. Figure 9 shows the MAPE of the prediction using SVM models with different input state vectors. Then, the data from Fenjiang Rd 2 are used to test the models' performance.

As shown in Figure 9, when the state vector numbers of the upstream road link and current road link

Table 2
Models and corresponding state vectors

	Upstream road link		*Target road link*		*Downstream road link*	
	$S_{m-1}(t)$	$S_{m-1}(t–1)$	$S_m(t)$	$S_m(t–1)$	$S_{m+1}(t)$	$S_{m+1}(t–1)$
Model 1	√		√	√	√	
Model 2	√	√	√	√	√	
Model 3			√	√	√	√
Model 4	√		√	√	√	√
Model 5	√	√	√	√	√	√

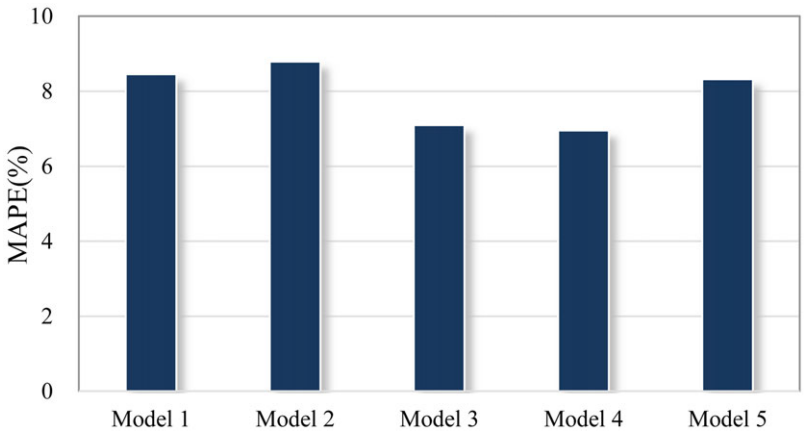


Fig. 9. MAPE values of SVM models with various state vectors.

are the same, the accuracy of Model 4 is higher than that of Model 1 and the accuracy of Model 5 is higher than that of Model 2, highlighting the importance of the downstream road link information for prediction accuracy. It is generally known that traffic speed changes rapidly. The state of the target road link during time t was influenced directly by the state of the upstream road link during time t. The state of the upstream road link during time t-1 had little influence on the state of the target road link during time t. The relationship between $S_{m-1}(t-1)$ and $S_m(t)$ was very weak. Prediction accuracy is degraded by excessive dependence on the upstream information. Hence, Model 5 is not more accurate than Model 4, and Model 2 is not more accurate than Model 1. Overall, Model 4 is the best and exhibited the lowest MAPE. Therefore, it is used in this article.

In addition, the radial basis function (RBF) kernel function is used for the SVM model in this study (Yu et al., 2010, 2011). Before applying the SVM, two parameters, C and e, are first determined. In identifying the parameters in SVM models with different input state vectors, grid-search is used to identify the optimal parameter values. Because the MAPE of Model 4 is the lowest for the SVM, this study adopted the values of the two parameters (C, e) as (1.57, 0.03).

6.2 Short-term traffic speed prediction

The short-term traffic speed prediction model involves the subsequent road links, and a taxi arriving at one of these road links is a future event. Considering the satisfactory performance of the 4th SVM model in this article, a short-term traffic speed prediction model containing road links from the origin (road link 1) to the destination (road link 5) is tested based on the prediction results of multiple single-step predictions by this model. Figure 10 shows the prediction errors of traffic speed from the origin to the destination. Furthermore, the short-term traffic speed prediction model (dynamic) was compared with traditional traffic speed prediction model (static). From the comparison, the former performs better than the latter.

As shown in Figure 10, the MAPE of the short-term traffic speed prediction model varies from 3.31% to 15.35% on a route with a total length of 7.7 km. We find that the errors of the short-term traffic speed prediction model do not tend to increase with time. However, it can be observed that the prediction errors are small before 7:45 and after 8:25 (the off-peak period) compared with those between 7:45 and 8:25 (the peak period). In the off-peak periods, the driving time of the corridor can

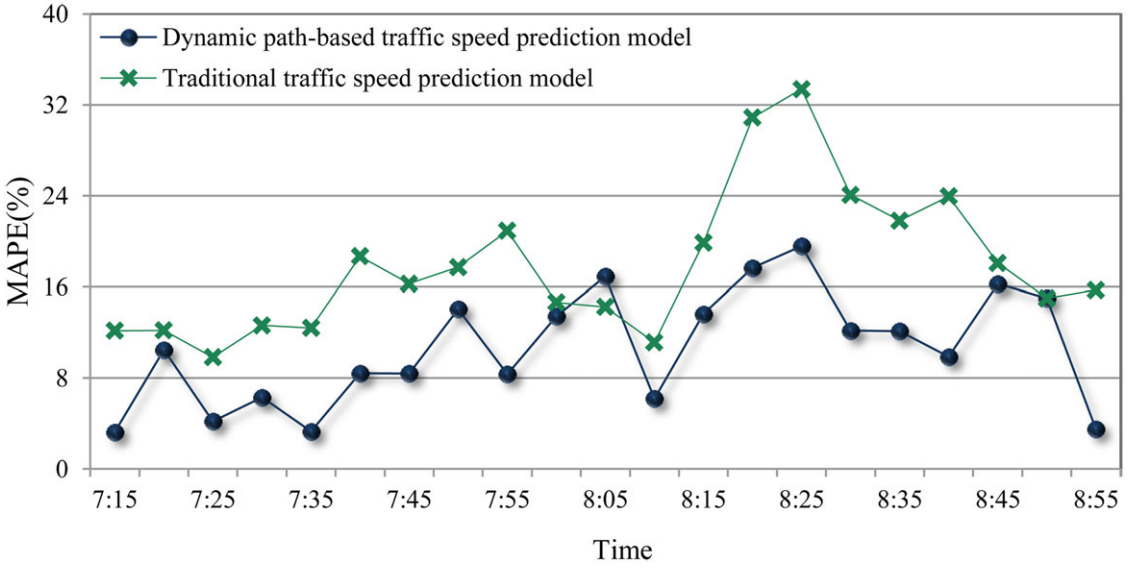


Fig. 10. Comparisons between short-term traffic speed prediction model and traditional traffic speed prediction model.

Table 3
MAPEs of SVM and ANN models with different state vectors

State vectors	*MAPE of SVM*	*MAPE of ANN*
Model 1	8.45	7.12
Model 2	8.79	8.82
Model 3	7.09	8.11
Model 4	7.05	7.64
Model 5	8.32	8.97

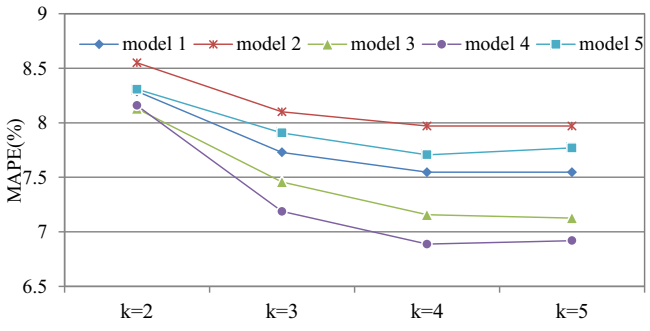


Fig. 11. MAPEs of *k*-NN models with different parameters.

be estimated using a few-step prediction for each road link. For example, if we want to predict the driving time from the entry of road link 1 to the exit of road link 5, the driving time from the entry of road link 1 to the exit of road link 5 would be more than one time interval (5 minutes) at the relatively low traffic speed. When the driving time is beyond several time intervals, the single-step model is no longer suitable for prediction. Thus, the single-step prediction model is used to predict traffic conditions on the target road link at the next interval. The short-term traffic speed prediction model used in this article is an extension of the single-step prediction model. However, the predicted data calculated from the previous steps (single-step prediction model) were used successively as the input data in the following steps (short-term traffic prediction model). The errors generated from the predicted data gradually accumulated with additional input data for the multi-step prediction models, especially during the peak period between 7:45 and 8:25. Therefore, the accumulated prediction error in the peak period exceeded that in the off-peak period. From Figure 10, it can be observed that in the peak period between 7:45 and 8:25, the prediction errors are far higher than those of the off-peak period, except at two points.

6.3 Prediction performance

To evaluate the model prediction performance of the short-term traffic speed prediction model, we compare the short-term traffic speed prediction model with the moving average data-based model, historical data-based model, ANN model, and *k*-NN model.

The formula of the moving average data-based model is shown in Equation (14).

$$\hat{S}_m(t) = \frac{1}{r}\{S_m(t-1) + S_m(t-2) + S_m(t-3) + \cdots + S_m(t-r)\} \quad (14)$$

Table 4
Value of (*p, d, q*) for each road

Road	*Fenjiang Rd 1*	*Fenjiang Rd 2*	*Fenjiang Rd 3*	*Fenjiang Rd 4*	*Fenjiang Rd 5*	*Fenjiang Rd 6*
(*p, d, q*)	(4, 1, 1)	(2, 1, 2)	(1, 1, 1)	(2,1, 1)	(1, 1, 1)	(2, 1, 3)

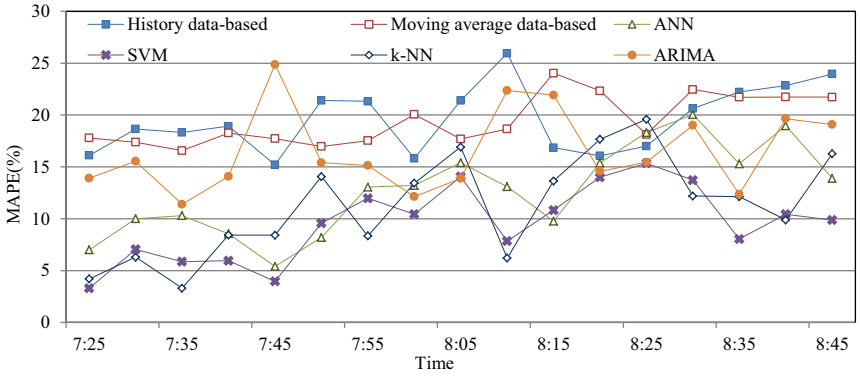


Fig. 12. Prediction accuracy comparison of six models.

where r is the number of previous time periods used. The travel speed in time period t on road link m is predicted by the speed in the previous time period. The data used for prediction is the previous data of the same day. After the numerical example test, r is set to 3 because this value provided the best prediction accuracy.

The formula of the historical data-based model is shown in Equation (15).

$$\hat{S}_m(t) = \frac{1}{h}\sum_{i=1}^{P} S_{mi}(t) \tag{15}$$

where $\hat{S}_m(t)$ is the average speed of all taxis during the time period t in the historical database, $S_{mi}(t)$ is the average speed of the taxis during the same time interval t of day i in the historical database, and h is the total days in the historical database.

To determine the inputs for the ANN model, sensitivity tests were conducted. The single-time-step model was used to calibrate the two models. The input state vectors of the ANN models are the same as those of the SVM model (shown in Table 2), and the data used to calibrate the ANN models are the same as those of the SVM model. Similar to the SVM model, the input state vectors of a standard three-layer ANN model are determined based on the results of sensitivity tests. The input state vectors of the ANN models are the same as those of the sensitivity tests of the SVM model. ANN models with different input state vectors have also been trained by the BP algorithm, and the results are shown in Table 3. In contrast to the SVM, the best input state vectors of the ANN are the state vector combination of Model 1. After determining the inputs of the ANN model, a scaled conjugate gradient algorithm (Moller, 1993) is used to train the ANN model. The number of hidden neurons is determined to be six in this study.

In the k-NN model with different numbers of nearest neighbors (k), the MAPE of the model with $k = 2$ among the four studied models is the highest (Figure 11). Therefore, the two nearest neighbors searched cannot comprehensively explain all the test data. When $k = 3$, the MAPEs decline rapidly, and the prediction accuracy increases. When k exceeds 3, the overall volatility is not large, and when $k \geq 4$, the volatility is small. Comparing the MAPEs with different values of k reveals that the prediction errors for road link 2 are lowest and are similar when $k = 4$ or 5. Thus, the value of k is set to 4 in this study. In addition, the ARIMA is adopted here. Table 4 shows the parameters (*p, d, q*).

The prediction accuracies are compared in Figure 12. The results show that the ARIMA does not obtain an ideal result. The average MAPEs of the historical data-based model and moving average data-based model exceed 15%, and the average MAPEs of the SVM model, ANN model, and k-NN model are relatively low and comparable. The results predicted by SVM and ANN exhibit the same trend because of their similar fundamental construction. However, the performance of the SVM model is better than that of the ANN model because the ANN model can more easily converge to a local maximum. When comparing the SVM model and k-NN model, the k-NN model clearly exhibits larger fluctuations, indicating that the SVM model is more

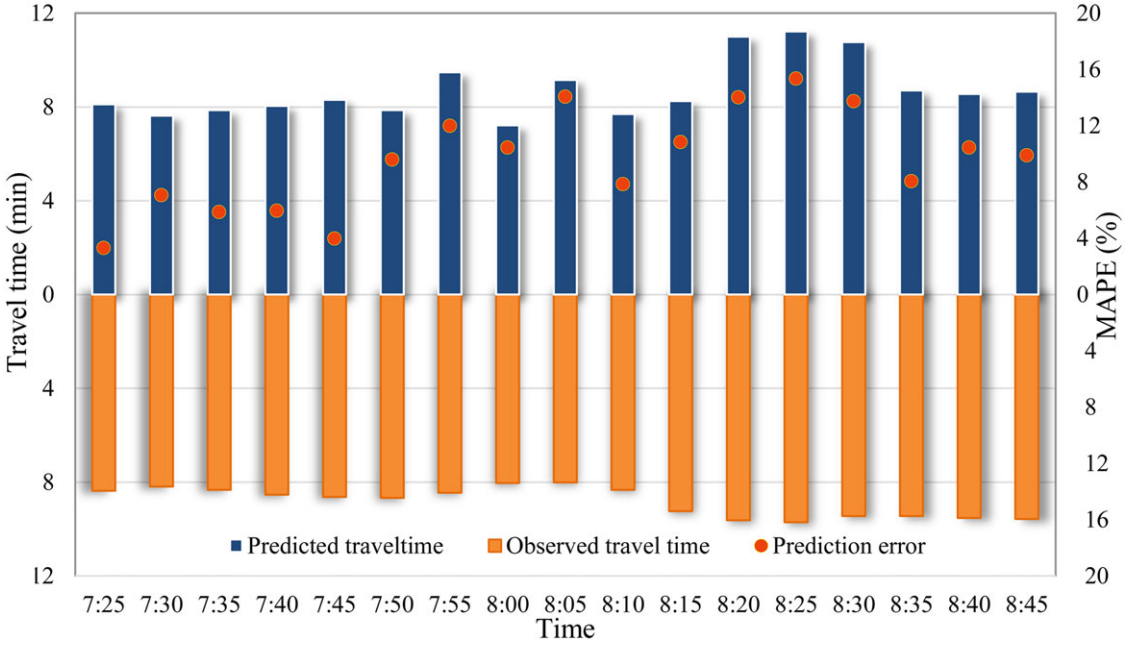


Fig. 13. Comparisons between observed and predicted travel times.

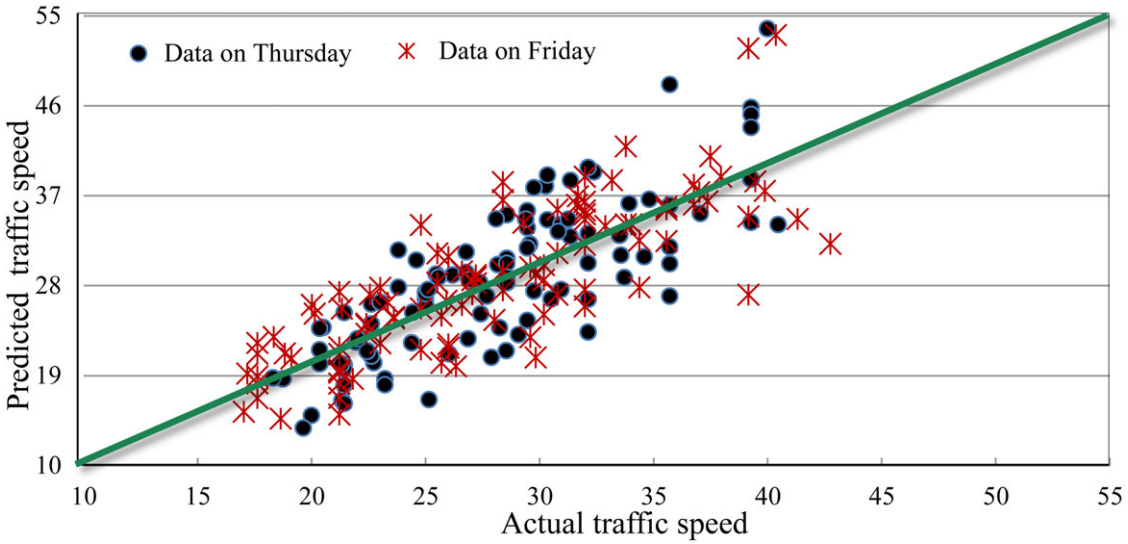


Fig. 14. Predictability of the model on traffic speed.

effective. In terms of time, we find that in the off-peak period (before 7:45 and after 8:25), the traffic speeds are similar to historical data and that the accuracy of the SVM model is better than or the same as those of the ANN and *k*-NN models. However, in the morning peak period (between 7:45 and 8:25), the accuracy of all models decreased because of the relatively complex traffic environment. Overall, the SVM model provides good prediction accuracy. Based on the results, the SVM model is a powerful tool for short-term traffic speed prediction. Furthermore, the time series of the observed and predicted travel times are shown in Figure 13. The results show that the error of predicted travel time varies from 4% to 16%. The average MAPE of the SVM model is less than 16%. In Figure 14, the predicted traffic speed is compared to the actual traffic speed. The validation test results, reported in Figure 14, demonstrate

that the short-term prediction mode provides good performance in predicting traffic speed. This model offers satisfactory accuracy for traffic speeds lower than 35 km/h. However, when the traffic speed is greater than 35 km/h, the prediction accuracy is reduced. According to the analysis, the main reason is that the data collection accuracy is degraded when the traffic speed is higher.

7 CONCLUSIONS

The goal of this work is to devise an accurate method for short-term traffic prediction and thereby support travellers' route choices and traffic guidance/control. Based on the SVM algorithm, a novel single-time-step prediction model that synthetically considers spatial and temporal parameters is developed. A short-term traffic speed prediction model is then constructed based on the single-step prediction model.

In a test case using actual data, GPS information from taxis in Foshan city was collected to validate the performance of the proposed method. Six road links on Fenjiang Road were taken as a test area to validate the performance of the proposed methods, and the time interval for prediction was set to 5 min. Although the small interval of the 5-min data segmentation can mitigate the influence of traffic lights and intersections to some degree, the short-term traffic speed on an arterial road remains challenging to predict because of its complex characteristics. The short-term traffic speed prediction model involves subsequent road links, and a taxi arriving at one of these road links is considered as a future event. Moreover, compared with other models, the SVM model with spatial-temporal parameters offers better performance than the historical data-based model, moving average data-based model, ANN model, and k-NN model. Overall, despite the sophisticated road conditions in the empirical test, our model obtains desirable results with low MAPEs, and the results remain relatively satisfactory, even for multi-step prediction. Therefore, the SVM-based model is confirmed to be capable of predicting short-term traffic speed. The proposed short-term traffic speed prediction model can thus provide more comprehensive traffic guidance for administrators and travellers.

8 LIST OF SYMBOLS

m	mth target road link
t	time period of vehicle entering within mth road link
t^i	ith time period
$S_m(t)$	actual speed on mth road link in time period t
$\hat{S}_m(t)$	prediction speed of vehicle on mth road link in time period t
Φ^{T}	temporal state vector of the target road link
Φ^{U}	traffic speed of the upstream road links
Φ^{D}	traffic speed of the downstream road links
Γ	travel time from beginning to end
M	number of the links between the origin and the destination time interval (The time interval used in the article is 5 min)
T_1	time of vehicle arrival at 1st road link
T_m	time of vehicle arrival at mth road link
$\hat{T}_m$	travel time of vehicle entering within mth road link
L_m	length of mth road link
$\hat{S}_m(t)$	prediction speed of vehicle in time period t on mth road link
d	spherical distance
r	radius of the earth
x_1	dimension of A
y_1	precision of A
x_2	dimension of D
y_2	precision of D
N	number of times that taxi sends GPS data on road link m
$V_i(t)$	speed of taxi at each position
P	number of prediction
r	number of previous time periods used
$S_{mi}(t)$	average speed of the taxis during the same time interval t of a day i in the historical database
h	total days in the historical database

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China 51578112 and 71571026, National Basic Research Program of China 2012CB725401, the Trans-Century Training Program Foundation for Talents from the Ministry of Education of China NCET-12-0752 and the Fundamental Research Funds for the Central Universities 3132015062 and DUT16YQ104.

REFERENCES

Adeli, H. (2001), Neural networks in civil engineering: 1989–2000, *Computer-Aided Civil and Infrastructure Engineering*, **16**(2), 126–42.

Adeli, H. & Jiang, X. (2003), Neuro-fuzzy logic model for freeway work zone capacity estimation, *Journal of Transportation Engineering*, **129**(5), 484–93.

Adeli, H. & Karim, A. (2000), Fuzzy-wavelet RBFNN model for freeway incident detection, *Journal of Transportation Engineering*, **126**(6), 464–71.

Akbari, M., van Overloop, P. J. & Afshar, A. (2011), Clustered k nearest neighbor algorithm for daily inflow forecasting, *Water Resource Management*, **25**(5), 1341–57.

Antoniou, C., Ben-Akiva, M. & Koutsopoulos, H. N. (2007), Nonlinear Kalman filtering algorithms for on-line calibration of dynamic traffic assignment models, *IEEE Transactions on Intelligent Transportation Systems*, **8**(4), 661–70.

Boto-Giralda, D., Diaz-Pernas, F. J., Gonzalez-Ortega, D., Diez-Higuera, J. F., Anton-Rodriguez, M. & Martinez-Zarzuela, M. (2010), Wavelet-based denoising for traffic volume time series forecasting with self-organizing neural networks, *Computer-Aided Civil and Infrastructure Engineering*, **25**(7), 530–45.

Chang, H., Lee, Y., Yoon, B. & Baek, S. (2012), Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences, *Intelligent Transport Systems, IET*, **6**(3), 292–305.

Chen, G., Govindan, K. & Golias, M. M. (2013), Reducing truck emissions at container terminals in a low carbon economy: proposal of a queueing-based bi-objective model for optimizing truck arrival pattern, *Transportation Research Part E: Logistics and Transportation Review*, **55**, 3–22.

Chen, G., Govindan, K. & Yang, Z. Z. (2012), Managing truck arrivals with time windows to alleviate gate congestion at container terminals, *International Journal of Production Economics*, **141**(1), 179–88.

Chen, G. & Yang, Z. Z. (2010), Optimizing time windows for managing arrivals of export container in Chinese container terminals, *Maritime Economics & Logistics*, **12**(1), 111–26.

Clark, S. (2003), Traffic prediction using multivariate nonparametric regression, *Journal of Transportation Engineering*, **129**(2), 161–68.

Dharia, A. & Adeli, H. (2003), Neural network model for rapid forecasting of freeway link travel time, *Engineering Applications of Artificial Intelligence*, **16**(7), 607–13.

Dia, H. (2001), An object-oriented neural network approach to short-term traffic forecasting, *European Journal of Operational Research*, **131**(2), 253–61.

Dougherty, M. S. & Cobett, M. R. (1997), Short-term inter-urban traffic forecasts using neural networks, *International Journal of Forecasting*, **13**(1), 21–31.

Ghosh-Dastidar, S. & Adeli, H. (2003), Wavelet-clustering-neural network model for freeway incident detection, *Computer-Aided Civil and Infrastructure Engineering*, **18**(5), 325–38.

Ghosh-Dastidar S. & Adeli H. (2006), Neural network-wavelet microsimulation model for delay and queue length estimation at freeway work zones, *Journal of Transportation Engineering*, **132**(4), 331–41.

Hamed, M. M., Al-Masaeid, H. R. & Bani Said, Z. M. (1995), Short term prediction of traffic volume in urban arterials, *Journal of Transportation Engineering*, **121**(3), 249–54.

Hofleitner, A., Herring, R. & Bayen, A. (2012), Arterial travel time forecast with streaming data: a hybrid approach of flow modeling and machine learning, *Transportation Research Part B: Methodological*, **46**(9), 1097–1122.

Ishak, S., Kotha, P. & Alecsandru, C. (2003), Optimization of dynamic neural network performance for short-term traffic prediction, *Transportation Research Record*, **1836**(7), 45–56.

Jiang, X. & Adeli, H. (2003), Freeway work zone traffic delay and cost optimization model, *Journal of Transportation Engineering*, **129**(3), 230–41.

Jiang, X. & Adeli, H. (2004a), Object-oriented model for freeway work zone capacity and queue delay estimation, *Computer-Aided Civil and Infrastructure Engineering*, **19**(2), 144–56.

Jiang, X. & Adeli, H. (2004b), Wavelet packet-autocorrelation function method for traffic flow pattern analysis, *Computer-Aided Civil and Infrastructure Engineering*, **19**(5), 324–37.

Jiang, X. & Adeli, H. (2005), Dynamic wavelet neural network model for traffic flow forecasting, *Journal of Transportation Engineering*, **131**(10), 771–79.

Karim, A. & Adeli, H. (2002), Comparison of fuzzy-wavelet radial basis function neural network freeway incident detection model with California algorithm, *Journal of Transportation Engineering*, **28**(1), 21–30.

Karim, A. & Adeli, H. (2003a), Fast automatic incident detection on urban and rural freeways using wavelet energy algorithm, *Journal of Transportation Engineering*, **129**(1), 57–68.

Karim, A. & Adeli, H. (2003b), CBR model for freeway work zone traffic management, *Journal of Transportation Engineering*, **129**(2), 134–45.

Karim, A. & Adeli, H. (2003c), Radial basis function neural network for work zone capacity and queue estimation, *Journal of Transportation Engineering*, **129**(5), 494–503.

Kayacan, E., Ulutas, B. & Kaynak, O. (2010), Grey system theory-based models in time series prediction, *Expert Systems with Applications*, **37**(2), 1784–89.

Kirby, H. R., Watson, S. M. & Dougherty, M. S. (1997), Should we use neural network or statistical models for short-term motorway traffic forecasting, *International Journal of Forecasting*, **13**(1), 43–50.

Lam W. H. K., Tang, Y. F. & Tam, M.-L. (2006), Comparison of two non-parametric models for daily traffic forecasting in Hong Kong, *Journal of Forecasting*, **25**(3), 173–92.

Mallat, S. (1999), *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, CA.

Manoel, C. N., Jeong, Y. S., Jeong, M. K. & Han, L. D. (2009), Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions, *Expert Systems with Applications*, **36**(3), 6164–73.

Min, W. L. & Wynter, L. (2011), Real-time road traffic prediction with spatio-temporal correlations, *Transportation Research Part C: Emerging Technologies*, **19**(4), 606–16.

Moller, M. F. (1993), A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks*, **6**(4): 523–33.

Nagel, K., Esser, J. & Rickert, M. (2000), Large-scale traffic simulations for transportation planning, *Annual Review of Computational Physics*, **7**, 151–202.

Ojeda, L. L., Kibangou, A. Y. & De Wit, C. C. (2013), Adaptive Kalman filtering for multi-step ahead traffic flow prediction, in *Proceedings of the American Control Conference (ACC), IEEE*, Washington, USA, 4724–29.

Okutani, I. & Stephanedes, Y. J. (1984), Dynamic prediction of traffic volume through Kalman filtering theory, *Transportation Research Part B: Methodological*, **18**(1), 1–11.

Queen, C. M. & Albers, C. J. (2009), Intervention and causality: forecasting traffic flows using a dynamic Bayesian network, *Journal of the American Statistical Association*, **104**(486), 669–81.

Smith, B. L. & Demetsky, M. J. (1995), Short-term traffic flow prediction: neural network approach, *Transportation Research Record*, **1453**, 98–104.

Smith, B. L. & Demetsky, M. J. (1996), Multiple-interval freeway traffic flow forecasting, *Transportation Research Record*, **1554**(17), 136–41.

Smith, B. L. & Demetsky, M. J. (1997), Traffic flow forecasting: comparison of modeling approaches, *Journal of Transportation Engineering*, **123**(4), 261–66.

Smith, B. L., Wlliams, B. M. & Oswald, R. K., (2002), Comparison of parametric and nonparametric models for traffic flow forecasting, *Transportation Research Part C: Emerging Technologies*, **10**(4), 303–21.

Stathopoulos, A. & Karlaftis, M. G. (2003), A multivariate state-space approach to urban traffic flow prediction, *Transportation Research Part C: Emerging Technologies*, **11**(2), 121–35.

Tan, M. C., Wong, S. C., Xu, J. M., Guan, Z. R. & Zhang, P. (2009), An aggregation approach to short-term traffic flow prediction, *IEEE Transactions on Intelligent Transportation Systems*, **10**(1), 60–69.

Turochy, R. E. (2006), Enhancing short-term traffic forecasting with traffic condition information, *Journal of Transportation Engineering*, **132**(6), 469–74.

Vlahogianni, E. I., Golias, J. C. & Karlaftis, M. G. (2004), Short-term traffic forecasting: overview of objectives and methods, *Transport Reviews*, **24**(5), 533–57.

Vlahogianni, E. I., Karlaftis, M. G. & Golias, J. C. (2007), Spatio-temporal short-term urban traffic volume forecasting using genetically optimized modular networks, *Computer-Aided Civil and Infrastructure Engineering*, **22**(5), 317–25.

Vythoulkas, P. C. (1993), Alternative approaches to short-term traffic forecasting for use in driver information systems, *Transportation and Traffic Theory*, **12**, 485–506.

Wang, Y. & Papageorgiou, M. (2005), Real-time freeway traffic state estimation based on extended Kalman filter: a general approach, *Transportation Research Part B: Methodological*, **39**(2), 141–67.

Wei, Y. & Chen, M. C. (2012), Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks, *Transportation Research Part C: Emerging Technologies*, **21**(1), 148–62.

Williams, B. M. (2001), Multivariate vehicular traffic flow prediction: an evaluation of ARIMAX modeling, *Transportation Research Record: Journal of the Transportation Research Board*, **1776**, 194–200.

Williams, B. M., Durvasula, P. K. & Brown, D. E. (1998), Urban traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models, *Transportation Research Record: Journal of the Transportation Research Board*, **1644**, 132–44.

Wu, C. H., Ho, J. M. & Lee, D. T. (2004), Travel-time prediction with support vector regression, *IEEE Transactions on Intelligent Transportation Systems*, **5**(4), 276–81.

Yildirimoglu, M. & Geroliminis, N. (2013), Experienced travel time prediction for congested freeways, *Transportation Research Part B: Methodological*, **53**, 45–63.

Yu, B., Lam, W. H. & Tam, M. L. (2011), Bus arrival time prediction at bus stop with multiple routes. *Transportation Research Part C: Emerging Technologies*, **19**(6), 1157–70.

Yu, B., Yang, Z. Z., Chen, K. & Yu, B. (2010), Hybrid model for prediction of bus arrival times at next station, *Journal of Advanced Transportation*, **44**(3), 193–204.

Yu, B., Yang Z. Z. & Yao, B. Z. (2006), Bus arrival time prediction using support vector machines, *Journal of Intelligent Transportation Systems*, **10**(4): 151–58.

Zhang, L., Liu, Q., Yang, W., Wei, N. & Dong, D. (2013), An improved K-nearest neighbor model for short-term traffic flow prediction, *Procedia-Social and Behavioral Sciences*, **96**, 653–62.

Zhang, Y. D. & Wu, L. N. (2009), Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network, *Expert Systems with Applications*, **36**(5), 8849–54.

Zhang, Y. D. & Wu, L. N. (2012a), An MR brain images classifier via principal component analysis and kernel support vector machine, *Progress in Electromagnetics Research*, **130**, 369–88.

Zhang, Y. D. & Wu, L. N. (2012b), Classification of fruits using computer vision and a multi-class support vector machine, *Sensors*, **12**(9), 12489–505.

Zhang, Y. L. & Xie, Y. C. (2008), Forecasting of short-term freeway volume with v-support vector machines, *Transportation Research Record*, **2024**(11), 2–99.

Zheng, W. Z., Lee, D. H. & Shi, Q. X. (2006), Short-term freeway traffic flow prediction-Bayesian combined neural network approach, *Journal of Transportation Engineering*, **132**(2), 14–121.

Zuo, W. M., Zhang, D. & Wang, K. Q. (2008), On kernel difference-weighted k-nearest neighbor classification, *Pattern Analysis and Application*, **11**(3-4), 247–57.

(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Transportation Research Part A 46 (2012) 1265–1279

Contents lists available at SciVerse ScienceDirect

Transportation Research Part A

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

Real-time partway deadheading strategy based on transit service reliability assessment

Bin Yu *, Zhongzhen Yang, Shan Li

Transportation Management College, Dalian Maritime University, Dalian 116026, PR China

ARTICLE INFO

Article history:
Received 14 April 2011
Received in revised form 9 April 2012
Accepted 16 May 2012

Keywords:
Transit
Service reliability assessment
Partway deadheading strategy
Heuristic algorithm

ABSTRACT

This paper presents a partway deadheading strategy for transit operations to improve transit service of the peak directions of transit routes. This strategy consists of two phases: reliability assessment of further transit service and optimization of partway deadheading operation. The reliability assessment of further transit service, which is based on the current and recent service reliability, is used to justify whether or not to implement a partway deadheading operation. The objective of the second phase is to determine the beginning stop for a new service for the deadheaded vehicle by maximizing the benefit of transit system. A heuristic algorithm is also defined and implemented to estimate reliability of further transit service and to optimize partway deadheading operation. Then, the partway deadheading strategy proposed in this paper is tested with the data from a transit route in Dalian city of China. The results show the partway deadheading strategy with the reasonable parameters can improve transit service.

1. Introduction

Most cities in China are experiencing a rapid increase in motor ownership, which leads to a rising congestion, air pollution and high energy consumption. The developments of public transportation systems are given priority to solve or improve these urban transportation problems. However, transit service in most cities is still not satisfying due to various reasons. One of the most important factors influencing transit service is the unreliability of transit operation which will greatly discourage public transportation use. Transit operation is very complex due to some stochastic factors, such as weather, traffic incidents and interference from other traffic. A main task of transit system agency is to implement various control strategies to eliminate the influence of the disruptions in transit operation.

Generally, the scheduled headway at the peak direction is more difficult to be maintained compared to the reverse direction, especially at the peak periods. This can always induce bunching or large intervals between buses and the irregular loads (overloaded and almost empty buses) at the peak direction all the time. Fig. 1 shows the case of passengers waiting for bus at a stop of the peak direction of the transit route No. 23 in Dalian city. Although the headway of the route during peak period is 2.5 min, no bus arrives at the stop more than 20 min after the last bus departed. Due to the large passenger flow and traffic congestion, this case often occurs. To satisfy stranded passengers at stops, operators tend to add buses to at the peak direction with large passenger demand. Due to the limited number of the buses, one of the most efficient strategies is to dispatch a bus from the origin terminal of the off-peak direction to the origin terminal of the peak direction and restart a traffic service at the peak direction. However, the real-time operations (i.e. whether to dispatch the bus to the origin terminal of the peak direction and which bus to be dispatched) are both based on the operator's experience and might not improve transit

* Corresponding author.
E-mail address: minlfish@yahoo.com.cn (B. Yu).

0965-8564/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.tra.2012.05.009

Fig. 1. Passengers waiting for buses at a stop of the peak direction of the transit route No. 23.

operation situation timely and effectively. Therefore, there is a potential need to develop real-time control strategies to reduce the irregularity service at the peak direction.

Holding strategy is one of the most commonly real-time control strategies in transit operation. When a bus is ahead of the schedule, holding strategy is used to delay bus movement deliberately. Holding strategy can reduce headway variance and average waiting time of passengers. Holding strategy also increases travel time of passengers on board and the total bus cycle time.

Now intelligent transportation system (ITS) technologies have been applied into transit systems in many large or medium cities of China such as automatic bus location (AVL) or identification (AVI) systems and automatic passenger counters (APC), which provide the potential to remedy the disruptions of transit operation in real-time. Recently, in many studies on holding problems, researchers have assumed real-time information available. Eberlein et al. (2001) studied holding strategy with real-time information available. They constructed a mathematical programming model for the holding problem based on a rolling horizon scheme. They also developed a solution method for the model. The rolling horizon in their study can reduce the influence of the measure errors of real-time data. Dessouky et al. (2003) compared the performance of several holding strategies at a terminal. They also developed methods to forecast bus arrival times and passenger loads. They found the strategy with the most technologies can achieve the best performance. The strategy cannot only consider the time saved for late-arriving transfer passengers, but also concern with the delay for passengers who are already on-board, or will board at subsequent stops. van Oort et al. (2010) compared the schedule-based and headway-based holding strategies of short-frequency bus route. They also analyzed the impact of the maximum holding time on the performance of two holding strategies. Although holding strategies can efficiently improve the regularity of transit operation, too much slack in a schedule will reduce service frequency (Zhao et al., 2006).

Corresponding to holding strategy, station-skipping strategies, such as stop skipping, deadheading and short turning, run through some stops without service to reduce the total travel time and the headway between the bus and the preceding bus. Eberlein et al. (1999) discussed several real-time control strategies, such as deadheading, expressing, holding strategies and the combination of two among the strategies. They found that combined strategies are more efficient than any single strategy. Fu et al. (2003) obtained the similar results as the findings from Eberlein et al. (1999).

Eberlein et al. (1998) discussed the real-time deadheading strategy, in which a deadheaded vehicle runs empty from a terminal skipping several stops and starts its new service. The deadheading strategy needed to determine the dispatching time of the deadheaded vehicle and the beginning stop of the new service. They formulated the deadheading problem that minimized total passenger cost, and also developed a heuristic algorithm to solve the model. There are other studies (Furth, 1985; Ceder and Stern, 1981) on deadheading strategies.

Short-turning strategy is another dispatching control similar to deadheading strategy. The difference between the two strategies is that the beginning stop of the new service has to be determined in deadheading strategy, while short-turning point (the end point) of service needs to be decided in short-turning strategy. Some literature (Furth, 1987; Miller and Bunt, 1987; Vijayaraghavan and Anantharamaiah, 1995; Delle Site and Filippi, 1998) has optimized the location (stop) where a vehicle will be directed to turn around before the end of the route in a short-turning service. Some other researchers also discussed real-time short-turning strategies. Strathman et al. (2001) discussed several real-time control strategies including holding, short turning, and reassignment actions based on Tri-Met automated bus dispatching system. They analyzed the service regularity of transit operation and also mentioned when a bus should be chosen to turn around. Shen and Wilson (2001) developed a real-time disruption control model for rail transit systems, which includes holding, expressing and short-turning strategies. The paper did not include a methodology for short-turning strategy design. However, they found that the efficiency of short-turning operations was quite sensitive to the accuracy of the disruption duration estimate. In addition, some researchers (Strathman and Hopper, 1993; Kimpel et al., 2005) analyzed on-time performance of transit operation and presented several strategies, e.g., holding early arrivals (vehicles) and turning around late arrivals (vehicles) before it reaches the end of its route.

Although there have been many studies on station-skipping strategies, most researchers were mainly concerned with the construction and solution methods of the station-skipping problems. There is few literature to identify the conditions of

transit operation, i.e., justify whether transit service is reliable or not. The focus of this study is to develop a new deadheading strategy with transit service reliability assessment, which is called "partway deadheading strategy". According to this strategy, bus runs empty and chooses the fastest path, which is not necessarily the bus route path to complete the off-peak direction trip and restarts its service from some stop at the peak direction.

If the fleet size is held constant, this strategy is more suitable for the routes with the imbalance in demand between the two directions (e.g., the routes in most Chinese cities). Especially at peak periods, the strategy would be more effective. When a bus completes its operation during the peak direction (i.e., the bus reaches the destination terminal), the bus would not continue serve the reverse direction, but return to some stop and serve the peak-direction again. For the peak-direction, an additional bus in this strategy needs to be provided for reducing passengers' average waiting time. On the other hand, the strategy will require more cost for operating the additional bus and increase the passenger waiting time at the off-peak direction. Thus, it is important to determine the optimal beginning stop of the new service for the strategy. According to its practical application, once a disturbance occurs in traffic operation, it would require the real-time control strategy to remedy the unreliable transit service. In other word, service reliability in transit systems cannot be assured when a disturbance occurs, so real-time control strategy is intended to enhance the reliability of transit services.

Service reliability is one of the most important factors that are used to assess the operation situation of transit route. There is abundant literature on transit reliability assessment. The earliest researches on transit service reliability were made by Polus (1978) and Silcock (1981). Lin et al. (2008) presented a quality control framework to evaluate bus schedule adherence performances which were applied to quick and accurate quality control.

Chen et al. (2009) proposed bus service reliability analysis to improve the public transit service quality based on stop level, route level and network level respectively. There were three kinds of index used in their analysis, i.e. punctuality index based on routes, deviation index based on stops, and evenness index based on stops. Sorratini et al. (2008) assessed bus service reliability with micro-simulation by analyzing headway, excess waiting time, service regularity and recovery time of an urban network. And they proposed a public transport schemes to improve the reliability. Van Oort and van Nes (2009) presented a tool to estimate the influence of network changes on the bus route regularity and transit demand based on actual punctuality data of transit systems. Casello et al. (2009) presented a method to quantify the impacts of unreliable service on generalized passenger cost. Their results showed that the increase of reliability on bus arrivals can greatly decrease the generalized passenger cost.

The reliability of transit service is considered as one of the main factors influencing the degree of passengers' satisfaction. However, most researches on real-time control strategy have rarely considered the service reliability of routes, even though some studies have considered the on-time performance of buses at stops, e.g., holding strategy (Yu and Yang, 2009). In this study, we consider a transit route where transit service of the peak direction is always inadequate due to both large numbers of passengers and unreliable bus arrivals during the peak periods. Referring to the application in practice, the partway deadheading strategy proposed in this study is to decide when a bus should be deadheaded and which stop is the beginning stop for its new service at the peak direction of the deadheaded bus.

The reliability assessment of transit service of the route is firstly estimated according to average waiting time of passengers. Service reliability is used to determine whether the partway deadheading strategy is adopted. It may avoid error dispatching and improve the efficiency of the strategy. As the operation is unreliable, the partway deadheading strategy is optimized as a way to improve transit service. Then, the beginning stop for the new service of the deadheaded bus is determined by maximizing the benefit of the transit system. The study is organized as follows. The assessment of service reliability of the route and the optimization of the partway deadheading strategy are given in Section 2. Section 3 presents a heuristic algorithm. Section 4 reports computational results and the conclusions are discussed in Section 5.

2. Partway deadheading strategy development

2.1. Partway deadheading strategy description

A regular transit route with N stops is shown in Fig. 2. The terminals 1 (N) and $N/2$ ($N/2 + 1$) are the dispatching terminals, while the other stops are common ones. Assume the direction from terminal $N/2 + 1$ to terminal N is the peak direction with large passenger demand. The headway of two buses is the time interval arriving at the stop. $H_{i,j}$ denotes the headway between the successive buses i and $i - 1$ arriving at the stop j.

For simplification, we assume that partway deadheading strategy can only be applied to improve the transit service of the peak direction of the route, i.e., the buses at the origin terminal (i.e., terminal $N/2 + 1$) at the peak direction of the route cannot be permitted running empty and then provide transit service at the off-peak direction. In practice, it is common that the regular service of the peak direction is easier to be disturbed compared to the reverse direction, especially at peak periods. This can induce bunching or large intervals between buses. Irregular loads (overloaded and almost empty buses) at the peak direction always occur during peak periods. Therefore, there is a potential need to develop real-time control strategies to reduce the irregularity service at the peak direction for large cities in China.

In the partway deadheading strategy proposed in this study, the deadheaded bus does not provide regular transit service from the origin terminal (the terminal 1) at the off-peak direction. The deadheaded bus directly runs empty and restarts its service from the origin terminal or some stop of the peak direction (Fig. 2).

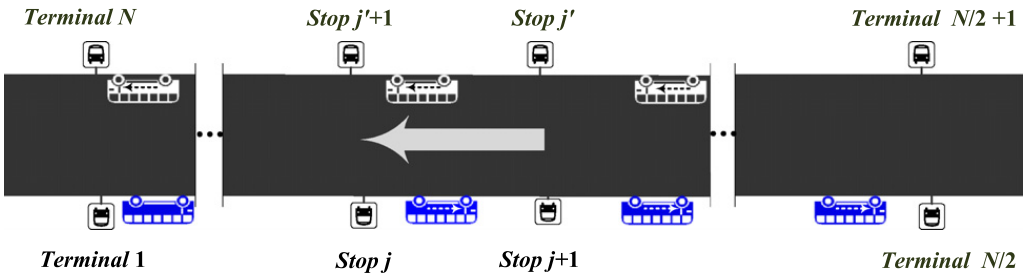


Fig. 2. An example bus route.

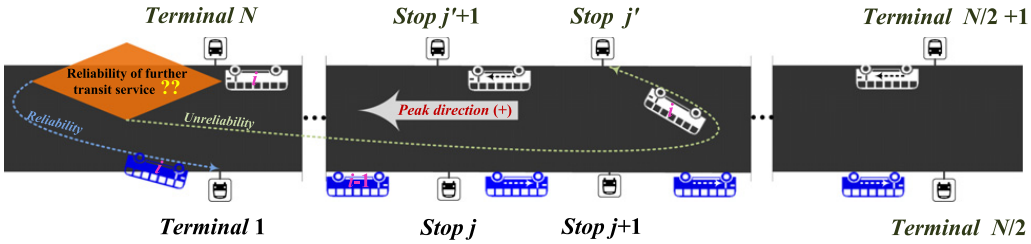


Fig. 3. Illustration of the partway deadheading strategy.

Since the deadheaded bus can choose the fastest path to its beginning stop for the new service, it consumes less time to join into transit service of the peak direction compared to the regular transit buses at the off-peak direction. As for the peak direction, the partway deadheading strategy can greatly decrease the waiting time of passengers at the stops after the beginning stop for the new service of the deadheading bus, as well as reduce the irregular service (overloaded and almost empty buses) and improve the service reliability of transit operation of the peak direction. However, it will increase the waiting time of passengers at the off-peak direction and the extra cost due to bus running empty.

The partway deadheading strategy (Fig. 3) proposed in this study consists of two steps: reliability assessment of further transit service of the route and optimization of the partway deadheading strategy.

When a bus has completed the operation of the peak direction, i.e., the bus arrives at the destination terminal of the peak direction, the reliability of current transit service of the route is first estimated. Then, the reliability of the further transit service of the route will be predicted. If the further transit service will be reliable, the target bus should not be deadheaded, but it should provide regular transit service at the off-peak direction. Otherwise, the target bus will be deadheaded and begin transit service from a stop of the peak direction. The beginning stop of the partway deadheading operation is determined by integrating saving waiting time of passengers at the peak direction, additional waiting time of passengers at the off-peak direction and extra cost of the partway deadheading running of the target bus. If the decided beginning stop is the terminal *N*, this indicates that the target bus should not be implemented partway deadheading operation, but provided regular transit service at the off-peak direction.

2.2. Reliability assessment of further transit service

Transit operation is always disrupted by some stochastic factors, especially at the peak period. Passengers are adversely affected by the consequences associated with unreliable service, such as additional waiting time and even switch alternative means of travel (such as car trip). Therefore, using waiting time of passengers to estimate the transit service reliability is a feasible way. Among the factors that may contribute to the variation of passenger waiting time, headways between buses are used to estimate the reliability of transit service.

2.2.1. Service reliability assessment of transit route

The irregular service (e.g., bunching or large headway) of transit route can increase average passenger waiting time. Thus, headways between bus arrivals at stops can indirectly represent transit service regularity. In this study, the square of the coefficient of variation of headways is used to estimate the reliability of transit service of a stop.

A rolling horizon of consecutive buses is introduced, which is considered as the effective information to assess the service irregularity of stops. That is, only the information of the buses in the rolling horizon is used to assess the service irregularity of the stop, while the information beyond the confine of the rolling horizon is skipped. Moreover, the horizon is rolled forward with buses running, e.g., if we assume the length of the rolling horizon is μ, as the bus υ arrives at the stop j, the information of the bus υ enters into the horizon, while the information of the bus $\upsilon-\mu$ is taken out of the horizon. Fig. 4 shows an example of how the horizon is rolled at the stop j as the bus υ arrives at the stop j.

$$\rho_j = 1 \Big/ \sqrt{\frac{\sigma_j^2}{\bar{h}_j}} \tag{1}$$

$$h_{i,j} = t_{i,j}^d - t_{i-1,j}^d \tag{2}$$

$$\bar{h}_j = \frac{\sum_{i\in\Omega_j} h_{i,j}}{\mu} \tag{3}$$

$$\sigma_j^2 = \frac{\sum_{i\in\Omega_j} [h_{i,j} - \bar{h}_j]^2}{\mu} \tag{4}$$

where $h_{i,j}$ denotes the actual headway between the bus i and $i-1$. $t_{i,j}^d, t_{i-1,j}^d$ denotes the actual departing time of the bus i and $i-1$ from the stop j. ρ_j denotes the reliability of transit service of a stop. Small values of the parameter indicate that transit service of the stop is unreliable, while large values mean that the headways of bus arrivals at the stop are relatively uniform.

When assessing the reliability of transit service of a bus route, the reliabilities of transit service of all the stops should be considered. To distinguish the contribution of the transit service reliability of each stop to the service reliability of the route, the number of passengers at each stop is considered as the relative weight to the service reliability of the entire route. The service reliability can also be defined as follow.

$$\Phi_i^{peak} = \sum_{j=N/2+1}^{N} \frac{p_j}{P} \rho_j \tag{5}$$

where Φ_i^{peak} denotes the current reliability of transit service with respect to the bus i. The higher Φ_i^{peak} is, the better the service reliability of the route shows. p_j is the number of passengers at the stop j. P is the total passengers at all the stops of the route.

2.2.2. Reliability prediction of further transit service

The bus operation is complex and the system is time-varying and unsteady. The current transit service is reliable or acceptable, but with the operation of the vehicles, the service level might drop substantially, i.e. the service of transit system becomes unreliable. If the measures are taken only based on the current service reliability, it will be difficult to improve the level of service in the short time. Therefore, if the unreliable service of the future buses can be predicted in advance and the corresponding scheduling strategy are adopted timely, it will be possible to avoid or reduce the impact of interference on transit operation.

When a bus arrives at the terminal (i.e., terminal N), the reliability of the current transit service can be assessed. Then, the reliability of the further transit service needs to be predicted. Similar to multiple step prediction of time series, the service reliability prediction of multiple time steps (i.e., multiple following buses) can be divided into direct and indirect categories (Cheng et al., 2008). Direct multiple step prediction constructs the independent models for each following bus, while indirect one uses the recursive method of single predictor. Generally, direct multiple step prediction models can provide better performance than indirect models. Therefore, the direct prediction model is used in this study.

To predict the reliability of the further transit service, the potential relation between the current and further transit services should be deduced. In this study, k-NN method is used to model the reliability of the further transit service based on the reliabilities of the current and recent transit services.

$$\hat{\Phi}_{i+\omega}^{peak} = f\left\{\Phi_i^{peak}, \ldots, \Phi_{i-\varepsilon+1}^{peak}\right\} \tag{6}$$

where $\hat{\Phi}_{i+\omega}^{peak}$ denotes the predicted reliability of the further $i+\omega$th bus. ω denotes the number of the prediction steps. ε is the number of input parameters of state judgment. $\Phi_i^{peak}, \ldots, \Phi_{i-\varepsilon+1}^{peak}$ represents the service reliability of the latest ε. f is the

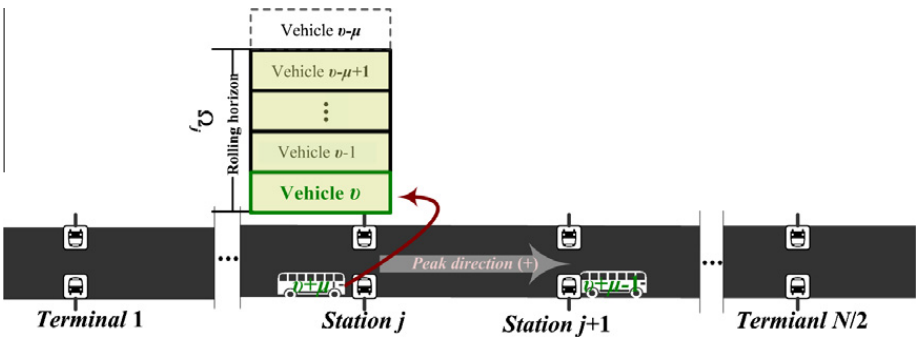


Fig. 4. Illustration of rolling horizon of buses.

function relation of the predicted service reliability and the service reliability of the latest ε. Fig. 5 shows an example of the further reliability of transit service ($\hat{\Phi}^{peak}_{i+\omega}$) is predicted as the bus i completes a regular service of the peak direction.

2.2.3. Applying k-NN in reliability prediction of transit service

The k-NN method is a relative old and simple method based on supervised learning (Bay, 1999; Yu et al., 2011). k-NN method is therefore much suitable for real-time applications due to its simple structure (without the need to estimate parameters). In the k-NN method for the reliability prediction of the further transit service, standard Euclidean distance is used to match the k nearest neighbors in the feature space. On the basis of Euclidean distance, the k nearest neighbors with the least distance to the input state can be determined. To weight the contributions of each neighbor, a common distance-based scheme is adopted to compute the weight of each neighbor. The forecasts can then be obtained by taking the weighted average of the observations from the k nearest neighbors.

$$\hat{\Phi}^{peak}_{i+\omega} = \sum_{j=1}^{k} \frac{1/d_j}{D} \times (\Phi^{peak}_{i'+\omega,j}) \tag{7}$$

$$d_j = \sqrt{\sum_{h=0}^{\varepsilon-1} (\Phi^{peak}_{i'-h,j} - \Phi^{peak}_{i-h})^2} \tag{8}$$

$$D = \sum_{j=1}^{k} \frac{1}{d_j} \tag{9}$$

where d_j represents the weighted distance between the jth nearest neighbor and the input state. $\Phi^{peak}_{i'+\omega,j}$ represents the service reliability with respect to the bus $i' + \omega$ in the rolling horizon of the jth nearest neighbor. D represents the sum of the weighted distance of the k nearest neighbors.

Fig. 6 shows an example of the k-NN method is applied to forecast the further reliability of transit service. Fig. 6a represents the foundational database, which is the basis for estimating whether the service of the future buses is reliable or not. The data which is stored in this database are historical measured reliability of transit service. The state vector $x_n = \{x^{(1)}, x^{(2)}, \ldots, x^{(\mu)}\}$ represents the current service reliability with respect to the bus in the rolling horizon (e.g. from the bus $i - \varepsilon + 1$ to the bus i in Fig. 5). When a new input state (e.g., x_m in Fig. 6b) appears, the k nearest neighbors (e.g., x_2 and x_3) are sought based on utilizing Euclidean distance. Then, the predicted value of the reliability of future transit service is given based on the formula (7).

If the predicted reliability of further transit service is lower than a given threshold (ϕ), it means that the transit service of the peak direction of the route can be disturbed by some factors. This will increase average waiting time of passengers of the peak direction without some control strategies. Under the conditions, a partway deadheading operation will be implemented to improve the further transit service. Since a partway deadheading operation shows the effort over some time, successive partway deadheading operations are not encouraged. In general, successive partway deadheading operations can achieve slightly more effort than a single partway deadheading operation due to their duplication of functions. Furthermore, successive partway deadheading operations also bring more disruption to the transit service at the off-peak direction. Thus, a coefficient λ, which is related to the number of the partway deadheading operations that do not begin their service, is introduced to prevent successive partway deadheading operations. However, when further transit service will be greatly more unreliable, successive partway deadheading operations could provide more efficient to improve the further transit service.

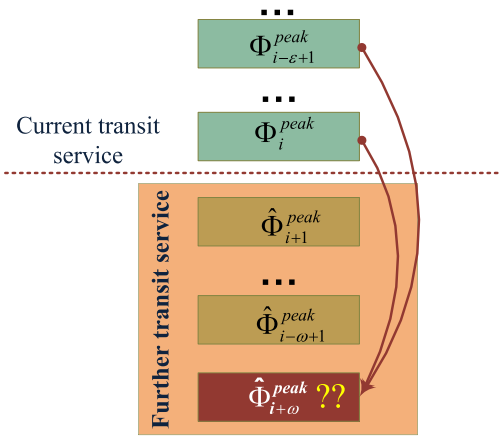


Fig. 5. Reliability prediction of further transit service.

$$\widehat{\Phi}_{i+\omega}^{peak} < \lambda \times \phi \tag{10}$$

where ϕ denotes the threshold of the reliability of further transit service. λ denotes the coefficient to control successive partway deadheading operations. In this study, λ is the reciprocal of the number of the deadheaded buses that has not reached the beginning stop of the new service, i.e., $0 < \lambda \leqslant 1$.

2.3. Partway deadheading formulation

As service of the route is unreliable and a partway deadheading strategy is to be implemented, the beginning stop for the new service should be determined. Partway deadheading strategy can reduce average waiting time of passengers of the peak direction, while it increases passenger waiting time of the reverse direction and extra running cost of the deadheaded bus. Therefore, the optimization of partway deadheading strategy should integrate the benefit of passengers of the peak and off-peak directions and extra running cost of the deadheaded bus to determine the beginning stop of the new service of the partway deadheading operation.

The definition of variables used throughout the model formulation is as follow:

2.3.1. Definition of variables

M	The total number of buses.
N	The total number of stops.
VC	Rated capacity of a standard bus.
r_j	Average passenger arrival rate at the stop j. Assume that passengers arrive randomly at a constant rate r_j at the stop j during peak periods.
q_j	Passenger alighting proportion at the stop j, which is a fixed constant computed by history data for the stop.
$\widehat{B}_{i,j}$	The number of the passengers boarding the bus i at the stop j.
$\widehat{A}_{i,j}$	The number of the passengers alighting from the bus i at the stop j.
$\widehat{L}_{i,j}$	Departure load of the bus i from the stop j.
$\widehat{R}_{i,j}$	The number of the passengers who are left by the bus i at the stop j and have to wait for the bus i + 1.
$\hat{t}^a_{i,j}$	Arrival time of the bus i at the stop j.
$\hat{t}^s_{i,j}$	Dwell times of the bus i at the stop j.
$\overline{t}^r_{j-1\to j}$	Average running time between the stop j − 1 and j.
$t^d_{i,j}$	Departure time of the bus i from the stop j.
$\hat{h}_{i,j}$	Headway between the buses i and i − 1 at the stop j, i.e., $\hat{h}_{i,j} = \hat{t}^d_{i,j} - \hat{t}^d_{i-1,j}$.
$\overline{T}^{shortest}_{\theta}$	The shortest running time from the destination terminal to the stop θ at the peak direction (the shortest running time can be computed with history data).
C^w	Unit time values associated with waiting time of passenger.
C^r	Unit cost associated with running time of bus.

2.3.2. Relation between passengers and bus operation

Assume running time between two adjacent stops can be achieved from history data. Thus, the arrival time of the bus *i* at the stop *j* can be yielded as follow:

$$\hat{t}^a_{i,j} = \hat{t}^d_{ij-1} + \overline{t}^r_{j-1\to j} \tag{11}$$

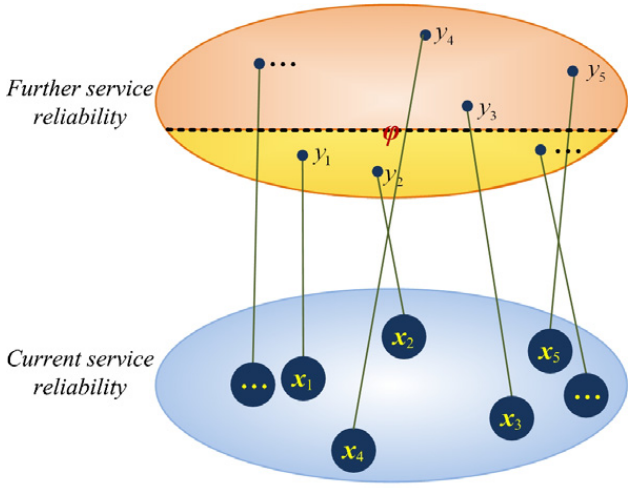
When computing the departure time of the bus *i* from the stop *j*, the dwelling time of the bus at the stop should be estimated first. Dwelling time of a bus is determined by the number of boarding or alighting passengers at the stop. The alighting passenger of the bus *i* at the stop *j* can be computed as follow:

$$\widehat{A}_{i,j} = q_j \times \widehat{L}_{i,j-1} \tag{12}$$

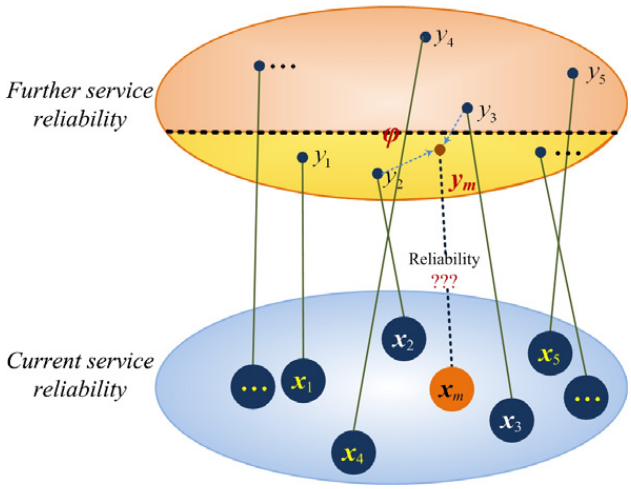
The boarding passengers include three parts: the passengers $\widehat{R}_{i-1,j}$ left by the bus $i-1$, the passengers $\hat{g}^-_{i,j}$ arriving from the bus $i-1$ departing to the bus *i* arriving and the passengers $\hat{g}^+_{i,j}$ arriving during the dwelling time of the bus at the stop. Thus, as the bus *i* arriving, the number of the passengers who expect to board the bus is equal to $(\hat{g}^-_{i,j} + \widehat{R}_{i-1,j})$ at the stop *j*.

$$\hat{g}^-_{i,j} = r_j \times (\hat{t}^a_{i,j} - \hat{t}^d_{i-1,j}) \tag{13}$$

Assume that all boarding takes place at the front door and alighting takes place at the rear door. The estimated dwelling time $\hat{t}^{ud}_{i,j}$ for passenger boarding and alighting time is equal to the longer one between the total boarding time and the total alighting time. That is,



(a) History service reliability database



(b) Reliability prediction using the k nearest neighbors

Fig. 6. The reliability prediction based on k-NN method. (a) History service reliability database and (b) reliability prediction using the k nearest neighbors.

$$\hat{t}_{i,j}^{ud} = \max[\bar{u} \times (\hat{g}_{i,j}^{-} + \widehat{R}_{i-1,j}), \bar{d} \times \widehat{A}_{m,j}] \tag{14}$$

where $\bar{u}$ and $\bar{d}$ denote average times of per boarding and alighting passenger respectively.

Then, the number of the passengers $\hat{g}_{i,j}^{+}$ arriving during the bus i dwelling at the stop j can be formulated as follow.

$$\hat{g}_{i,j}^{+} = r_j \times \hat{t}_{i,j}^{ud} \tag{15}$$

Thus, according to the capacity of bus, the number of the boarding passengers $\widehat{B}_{i,j}$ can be computed as follow.

$$\widehat{B}_{i,j} = \begin{cases} \hat{g}_{i,j}^{-} + \widehat{R}_{i-1,j} + \hat{g}_{i,j}^{+} & \text{if } \hat{g}_{i,j}^{-} + \widehat{R}_{i-1,j} + \hat{g}_{i,j}^{+} \leqslant VC - (\widehat{L}_{i,j-1} - \widehat{A}_{i,j}) \\ VC - (\widehat{L}_{i,j-1} - \widehat{A}_{i,j}) & \text{otherwise} \end{cases} \tag{16}$$

Then, the load of the bus i departing from the stop j can be updated and the number of the left passengers by the bus can also be yielded as follow.

$$\widehat{L}_{i,j} = \widehat{L}_{i,j-1} + \widehat{B}_{i,j} - \widehat{A}_{i,j} \tag{17}$$

$$\widehat{R}_{i,j} = \max[\hat{g}_{i,j}^{-} + \widehat{R}_{i-1,j} + \hat{g}_{i,j}^{+} - \widehat{B}_{i,j-1}, 0] \tag{18}$$

After achieving the number of the boarding passengers, the dwelling time of the bus i at the stop j can also be determined.

$$\hat{t}_{i,j}^{s} = \max[\bar{u} \times \widehat{B}_{i,j}, \bar{d} \times \widehat{A}_{i,j}] \tag{19}$$

Thus, substituting the arrival time and dwelling time of the bus i at the stop j in the following equation, the departure time of the bus from the stop can be yielded.

$$\hat{t}^d_{i,j} = \hat{t}^a_{i,j} + \hat{t}^s_{i,j} \tag{20}$$

In addition, when a bus arrives at a stop and the preceding bus does not depart from it, we assume that the preceding bus departs from the stop immediately as the current arriving.

2.3.3. Estimation of the location of the deadheaded bus

Assume that the deadheaded bus begins transit service at the peak direction of the route from the stop θ. $\overline{T}^{shortest}_{\theta}$ is used to denote the shortest running time from the destination terminal (the terminal N) to the stop θ at the peak direction. Then, the arrival time of the deadheaded bus i at the beginning stop θ of the new transit service can be defined as follow.

$$\hat{t}'^a_{i,\theta} = \hat{t}^d_{i,N} + \overline{T}^{\text{shortest}}_{\theta} \tag{21}$$

where $\hat{t}'^a_{i,\theta}$ denotes the arrival time of the bus i from stop N to θ.

As the deadheaded bus begins the new service, the bus will be inserted between two successive buses (e.g., $v-1$ and v). The bus $v-1$ will be the latest bus that has reached or gone through the stop j, while the bus v will be the nearest bus that has not reached the stop (Fig. 7). Furthermore, the arrival times of the buses $v-1$ and v should satisfy the following constrains.

$$\hat{t}^a_{v,\theta} \le \hat{t}'^a_{i,\theta} < \hat{t}^a_{v-1,\theta} \tag{22}$$

2.3.4. Mathematical model

Partway deadheading strategy can reduce average passenger waiting time of the peak direction (denoted as the positive benefit), while it also increase the additional passenger waiting time of the reserves direction and the extra running cost of the deadheaded bus (denoted as negative benefit). Therefore, the objective of the optimization model for partway deadheading strategy is to achieve the trade-off between the benefits of two parts during the fleet service cycle. The fleet service cycle means each bus in the fleet has completed its service cycle (i.e., the service cycle of a bus means a bus starting from the original terminal 1, going through the destination terminal $N/2$ and returning to the original terminal N). To simplify the model, assume when a bus completes a service cycle, the no. of the bus is increased by M (e.g., when bus $i-1$ finishes its transit service at both direction and return to the terminal N, its bus No. is changed to $M+i-1$).

Assume the current bus is i, when the bus $M+i-1$ has returned to the original terminal N, it can be regarded that the fleet has completed a cycle. Thus, the objective of the optimization model integrates the saving time cost of the passengers of the whole route and the extra running cost of the deadheaded buses if the bus i was implemented the partway deadheading operation.

$$\max F = C^w \times (T^- - T^+) - C^r \times \overline{T}^{\text{shortest}}_{\theta} \tag{23}$$

where F denotes the total cost of transit system, T^- is the total waiting time of passengers if the partway deadheading operation will not be implemented, and T^+ is the total waiting time of passengers with the implementation of the partway deadheading operation. $C^r \times \overline{T}^{\text{shortest}}_{\theta}$ is the extra running cost of the deadheaded buses.

The total waiting time of passengers without the partway deadheading operation (T^-) is first computed. Then, when the partway deadheading operation is implemented, the total waiting time of passengers (T^+) is the time with respect to the new bus queue.

2.3.4.1. Total waiting time of passengers without the partway deadheading operation (T^-). If the partway deadheading operation will not be implemented, each bus of the entire fleet provides regular transit service at both directions. Assume the bus queue of the entire fleet is $\{i-1, i, i+1, \ldots, M+i-2\}$. Then, the total waiting time of the passengers waiting for the buses $i-1$ to $i+M-2$ can be defined as follow.

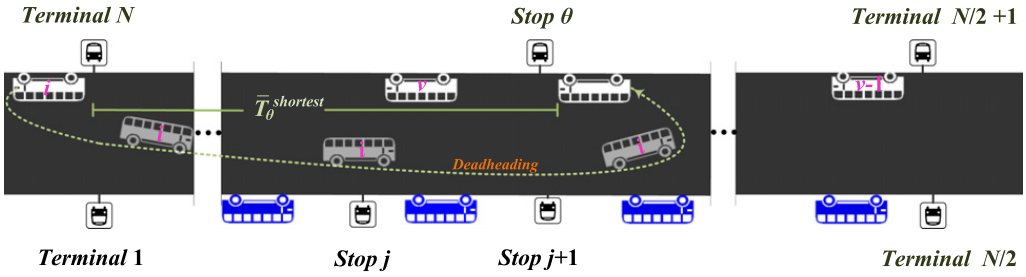


Fig. 7. An example of the location of the deadheaded bus.

$$T^{-} = \sum_{m=i-1}^{M+i-2} \sum_{j=1}^{N} \left(\widehat{B}_{m,j} \times \frac{\hat{h}_{m,j}}{2} + \widehat{R}_{m-1,j} \times \hat{h}_{m,j} \right) \tag{24}$$

2.3.4.2. Total waiting time of passengers with the partway deadheading operation (T^{+}). If implementing partway deadheading operation, as the deadheaded bus begins the new service, the bus queue will be varied. Assume the current vehicle is bus i, since the deadheaded bus i is inserted behind the bus $v-1$, the bus queue will be $\{i-1, i+1, \ldots, v, i, v+1, \ldots, M+i-2\}$. Let $i'-1$ be the first bus (i.e. the initial bus $i-1$), i' be the bus $i-1$, and do on, the bus queue can be renumbered as $\{i'-1, i', \ldots, M+i'-2\}$. Thus, the total waiting time of the passengers can be computed if implementing partway deadheading operation.

$$T^{+} = \sum_{m=i'-1}^{M+i'-2} \sum_{j=1}^{N} \left(\widehat{B}_{m,j} \times \frac{\hat{h}_{m,j}}{2} + \widehat{R}_{m-1,j} \times \hat{h}_{m,j} \right) \tag{25}$$

3. Solution algorithm

For the implementation of the partway deadheading strategy in the previous sections, a solution algorithm is presented to assess service reliability of the route and to determine the beginning stop of the new service for the partway deadheading strategy.

Step 1. Initialization
Determine the length of the rolling horizon (μ), the threshold value of service reliability assessment (ϕ, λ) of the route.
Step 2. Predict service reliability of the route
Step 2.1. Roll the horizon of the buses at each stop
As bus arrives at a stop, the oldest bus in the horizon is taken out and the current bus is inserted into the horizon.
Step 2.2. Compute the coefficient of variation of headways at each stop, according to Eq. (1).
Step 2.3. Compute the reliability of transit service Φ_i^{peak},, according to Eq. (5).
Step 2.4. Predict the reliability of the further transit service
if $\widehat{\Phi}_{i+\omega}^{peak} < \lambda \times \phi$goto **Step** 3; otherwise goto Step 5.
Step 3. Optimize the beginning stop for the new service of the partway deadheading strategy
Compute the average passenger arrival rate (r_j), average running time $(\overline{t}_{j-1-j}^{r})$ and the shortest running time from the destination terminal to each stop $(\overline{T}_{\theta}^{shortest})$.
Set $S^{No.} = 1$, where $S^{No.}$denotes the stop No.
Set $S^{optimal} = 0$, where $S^{optimal}$ denotes the current optimal beginning stop for the new service.
Set Max = 0, where Max denotes the objective function corresponding to $S^{optimal}$.
Step 3.1. Terminating optimization of the beginning stop for the new service.
If Set $S^{No.} > N/2$, go to Step 4; Otherwise go to Step 3.2.
Step 3.2. Estimate the location of the deadheaded bus, i.e., find the preceding bus $v-1$ and the following bus v of the deadheaded bus.
Step 3.3. Compute the number of the left passengers at the stop θ.
Step 3.4. Compute the benefit of the passengers at the peak direction due to the partway deadheading operation, according to Eq. (25).
Step 3.5. Compute the additional waiting time of the passengers at the off-peak direction, according to Eq. (28).
Step 3.6. Compute the extra time bus running empty.
Step 3.7. Compute the objective function, according to Eq. (29).
Step 3.8. If $F(S^{No.}) >$ Max, Max = $F(S^{No.})$ and $S^{optimal} = S^{No.}$.
Step 3.9. $S^{No.} = S^{No.} + 1$, goto Step 3.1.
Step 4. Implement the partway deadheading strategy with $S^{optimal}$.
Step 5. Terminating check of the partway deadheading strategy operating;
If exceeding the maximum time, then stop; Otherwise, goto Step 2.

4. Numerical test

The partway deadheading strategy proposed in this study is tested with the data of the route number 23 in Dalian city of China. The transit route No. 23 goes from suburb (the terminal 1) to city centre (the terminal 19) with total 19 stops and 14.5 km per direction (Fig. 8). The transit route is highly congested in the eastbound direction of the morning peak and

the direction is also determined as the peak direction in this study. The headway and the travel speed during peak period are 2.5 min and about 12–18 km/h. The details of the route are shown in Fig. 8.

The test period is during the morning peak (07:00–08:30) of typical weekdays in this study. The bus operational data, such as bus cycle times, passenger demands at stops, and traffic conditions on links, have been obtained from the analyzed route (Yu et al., 2012; Yang et al., 2007). The unit waiting time values of passenger and the unit running time cost of operator are set as 2.7 RMB/h and 1.5 RMB/h (Yu and Yang, 2009), respectively. The microscopic simulation model Paramics is applied to simulate bus operations, which is expected to provide a fairly reliable environment for testing the proposed partway deadheading strategies. The data during the morning peak of 30 weekdays were generated to test the proposed methods through the calibrated simulation model. Two third of the data were set to the history database used in the *k*-NN methods. The other data were set to the test data that was used to validate the performances of the prediction method of the service reliability and the partway deadheading strategy.

4.1. Model calibration

4.1.1. Length of the rolling horizon

Before the service reliability assessment of the route, the length of the rolling horizon of buses at stops should be determined, which can influence the accuracy of the assessment. To decide the rolling horizon, an examination is constructed based on the data of a typical day. Fig. 9 shows that the reliabilities of each bus at the peak direction of the typical day with various lengths of the rolling horizon.

It can be observed that the reliabilities of each transit service are fluctuated very badly, while the reliabilities are consistent with the increase of the length of rolling horizon. Especially, when the length of the rolling horizon is from 5 to 7, the reliabilities of each transit service are similar. This indicates that the increase of the length of the rolling horizon has little significance to the computation of service irregularity when $m = 5$.Thus, considering the computation time, the length of the rolling horizon is set to 5 in this study.

4.1.2. Parameters in the k-NN methods

k-NN methods have been documented while a large database is desirable for increasing the prediction accuracy. However, the large sample size has significant implications on the timeliness of model execution (Smith et al., 2002). Considering the timely feature of the real-time control strategies, the history database are divided into three parts according to each half hour, i.e., 07:00–07:30, 07:30–08:00 and 08:00–08:30. This can decrease the search space and greatly save computation time of the *k*-NN method. There are two parameters: the number of the input variables and the number of the nearest neighbor, in *k*-NN method. To calibrate the parameters, an examination, where the reliability of the next transit service will be predicted as each bus arriving at the terminal *N*, is constructed based on the data of a typical day. Fig. 10 shows the mean absolute percentage error (MAPE) of the service reliability prediction of the *k*-NN methods with various parameters.

It can be found that the parameter sets $\{\varepsilon = 4, k = 6\}$ and $\{\varepsilon = 5, k = 6\}$ are the best ones. Furthermore, the methods with more input variables (e.g., $\varepsilon = 4$ or 5) outperform the ones with less input variables (e.g., $\varepsilon = 2$ or 3). This can be attributed that when the number of the input parameters is small, it will be more difficult to identify the system status. In addition, we can find that the method with $k = 6$ is better than other methods for the service reliability prediction. Considering the prediction accuracy and computation time, the parameter set $\{\varepsilon = 4, k = 6\}$ is adopted for the service reliability prediction in this study.

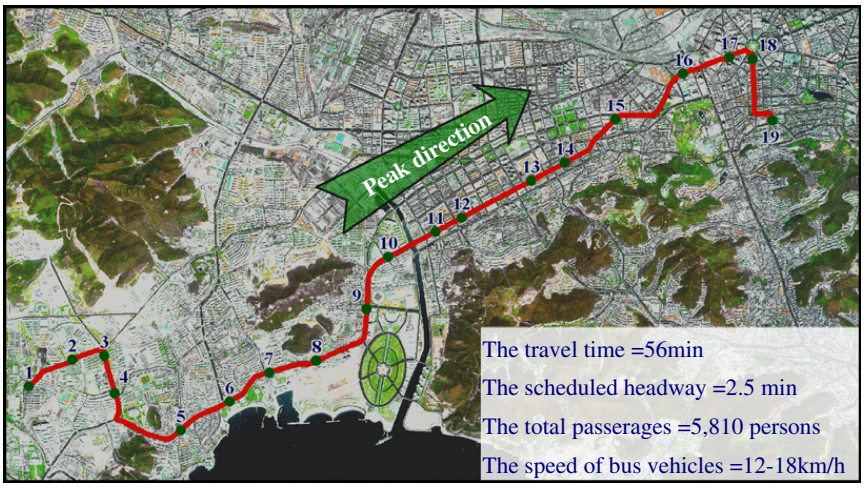


Fig. 8. Configurations of transit route 23 in Dalian city of China.

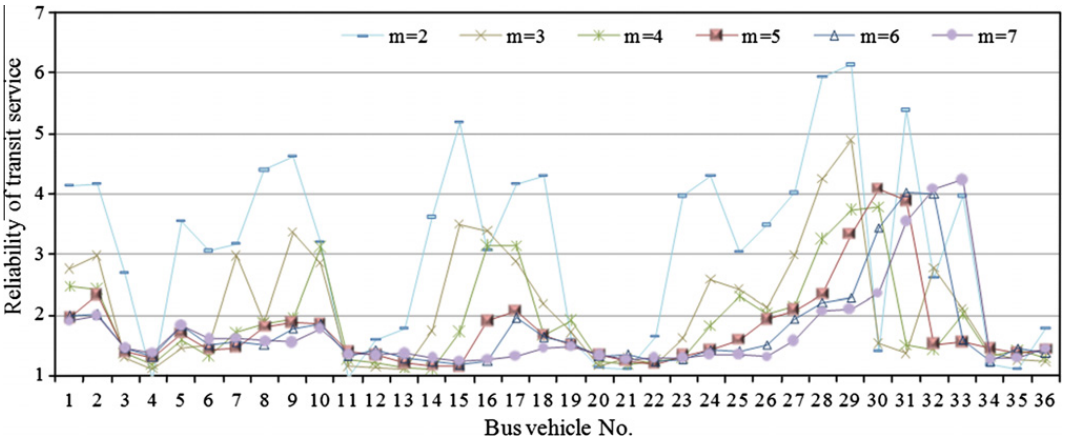


Fig. 9. The average service irregularity of the stop.

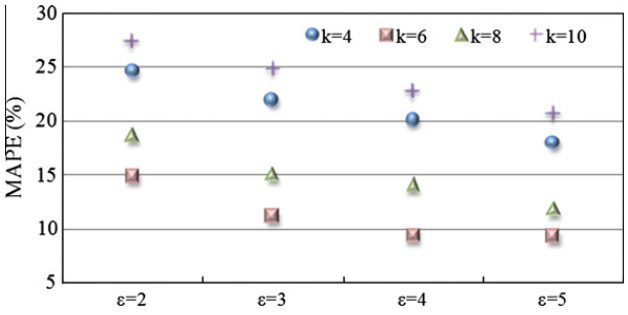


Fig. 10. The prediction results of the *k*-NN methods with various parameters.

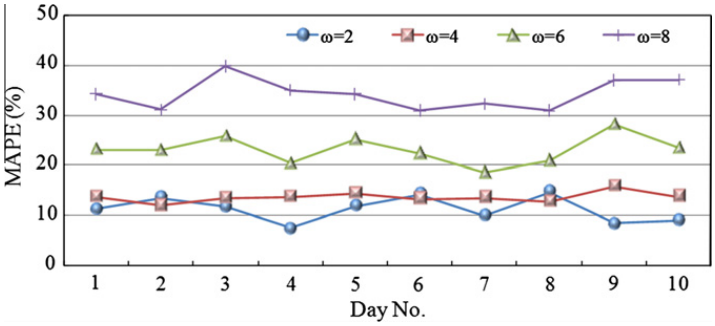


Fig. 11. The prediction results of the reliabilities with different transit service after the current service.

4.1.3. The number of time steps of prediction

Since the deadheaded bus needs to consume some time to arriving at the beginning stop of the new service, we expect to accurately predict the reliability of the further transit service, not just the reliability of the next transit service. Using the data to calibrate the parameters of *k*-NN method, the performances of the prediction model with various time steps are compared. Fig. 11 shows the prediction errors of the reliabilities of the second, forth, sixth and eighth transit service after the current service.

Obviously, the prediction error is larger with the increase of the number of the time steps, especially when $\omega = 6$ and $\omega = 8$. The prediction accuracy for the reliability of the next forth transit service is slightly worse than the one for the next second transit service. Thus, the predicted reliability of the forth transit service after the current service is used to justify whether the current bus should implement the partway deadheading operation or not.

4.1.4. Threshold value of service reliability of the route

The threshold ϕ can control the frequency of partway deadheading operations. Large threshold brings more operations, while small value reduces the frequency of the operations. To determine the suitable threshold value, the test data (the data of 10 days) are used to construct an examination. In the examination, the reliabilities of the transit service are analyzed if the total benefit of transit system is positive when the current bus implements the partway deadheading operation.

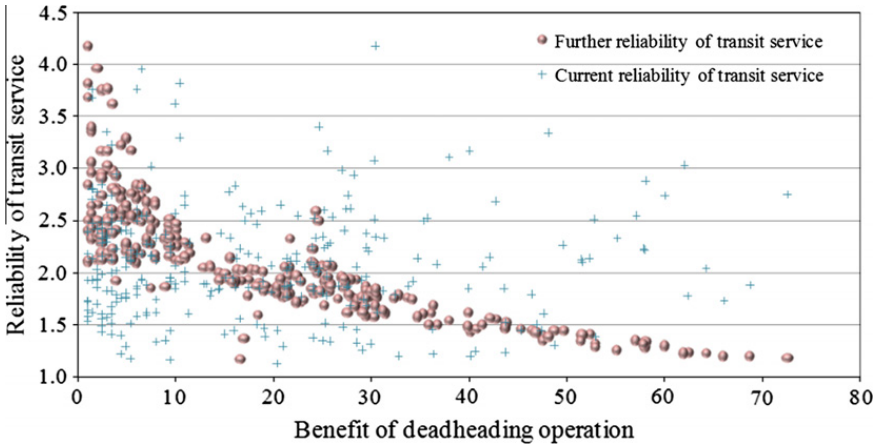


Fig. 12. The reliabilities of the current and predicted transit service.

Fig. 12 shows the benefit of each partway deadheading operation under the different current and further transit service reliability (i.e., the *i*th and *i* + 4th service) during the peak periods of 10 days, respectively.

From Fig. 12, we can find that the benefit of the partway deadheading operation is so random according to the current reliability of transit service. That is because the transit operation environment has the characteristics of time-varying and dynamic. Although the current operation status is subjected to little interference, the transit service in the future might turn to be serious and unreliable. Therefore, it may be difficult to obtain satisfactory results, if the operator decides whether to implement this partway deadheading strategy according to the current reliability. However, the benefit of partway deadheading operation shows higher correlations with the predicted reliability than with the current reliability. This indicates it can probably yield large benefit if justifying whether to implement the partway deadheading operation based on the future reliability of transit service. Furthermore, it can also be observed when the reliability is lower than 1.5, transit system can achieve more benefit. Thus, the threshold of service reliability of the route is set to 1.5.

4.2. Results

Without considering the reliability justification, Fig. 13 shows the total benefit of transit system and the beginning stop of the new service when each bus implements the partway deadheading operation based on the data of the typical day.

In general, the deadheaded bus can collect more passengers if it returns to the origin terminal at the peak direction and begins transit service. However, it can be observed that the beginning stops of most partway deadheading operations are not the original terminal, but the middle stops near the original terminal. This is because that the deadheaded bus needs to consume some running time to reach the beginning stop for the new service and if returning to the original terminal it needs to consume more running time. This can greatly increase the additional waiting cost of the passengers at the reverse direction and the extra running cost of the deadheaded bus. This indicates that the proposed model is efficient to integrate the benefit of the passengers in the two directions and the agency. In addition, during peak period, the implementation of partway deadheading operations for most buses except for the 1st, 2nd, 5th, 29th and 30th, can yield positive benefit for transit system due to larger passenger demand.

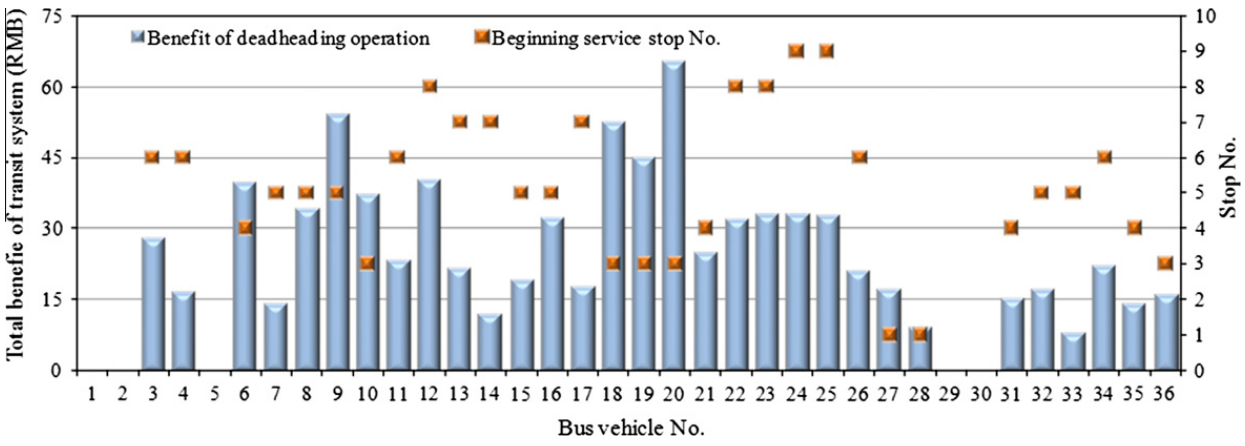


Fig. 13. The total benefit of transit system and the beginning stop with the partway deadheading.

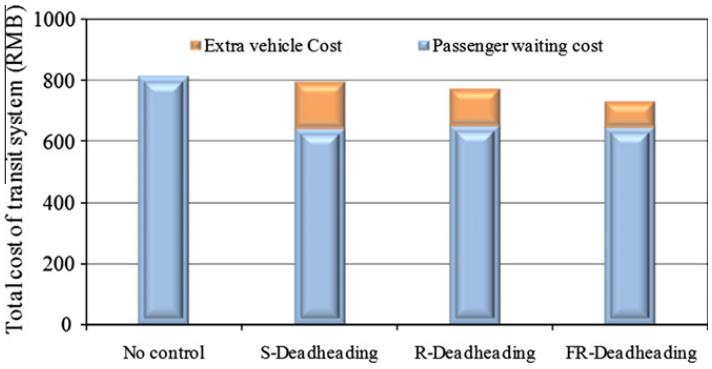


Fig. 14. The average total cost of transit system with four strategies.

Then, to evaluate the performance of the proposed strategy, no control (NC), a simple partway deadheading strategy (S-Deadheading), a partway deadheading strategy with current reliability-based justification (R-Deadheading) and the proposed strategy (FR-Deadheading) are constructed using the same data sets to the proposed hybrid model. In S-Deadheading, if the partway deadheading operation can bring positive benefit for transit system, the current bus will be implemented the partway deadheading operation. However, to avoid frequent partway deadheading operations, successive partway deadheading operations are not admitted. The difference between S-Deadheading and R-Deadheading is the base justifying whether to implement a partway deadheading operation or not. The former is based on the current reliability of transit service, while the later is based the predicted reliability. Other parameters in R-Deadheading are the same as the ones in FR-Deadheading. Under the same conditions, the four strategies are used to control the transit services of the test data. The results are evaluated in terms of the average total cost of transit system during 10 peak periods, which are shown in Fig. 14.

Obviously, the performance of the proposed strategy in this study is the best among the four strategies. The total cost using our strategy is less than the other strategies by 11.8%, 10.1%, 6.6% respectively. The performance of the NC strategy is the worst among four strategies due to no operation to remedy the unreliable service. As to the total cost, the performance of the R-Deadheading strategy is slightly better than the one of the S-Deadheading strategy.

Then, the performances of three partway deadheading strategies are compared. The S-Deadheading and FR-Deadheading strategies can bring less passenger cost than the R-Deadheading strategy. The extra bus running cost in the FR-Deadheading strategy is the least and the one in S-Deadheading strategy is the largest among three partway deadheading strategies. Considering the total cost in three deadheading strategies, it is obvious that there are some invalid deadheading operations in the S-Deadheading and R-Deadheading strategies, especially in the S-Deadheading strategy. This indicates that there is a high risk of error dispatching if not justifying transit service statement during real-time control. This can yield the expected effort of real-time control strategies and also waste the operation resource.

5. Conclusions

Our study has been motivated by real-life applications in China, where most transit routes have to serve large and imbalanced passengers. To improve transit service at the peak direction, it is a usual way that buses that should serve at the off-peak direction are transferred to the service at the peak direction.

In this study, a partway deadheading strategy was proposed to study when and how to dispatch the buses at the off-peak direction to the transit service of the peak direction of the route. Compared with other station-skipping strategies, the partway deadheading strategy can make full use of the resources of transit fleet, especially the buses serving at the off-peak direction. Therefore, this strategy is more effective to the routes which need to transport large and imbalanced passengers on both directions. In addition, this strategy is implemented at the origin terminal of the off-peak direction, so comparing to other stop-skipping strategies it can successfully avoid the inconvenience to the passengers on board. The partway deadheaded strategy is therefore much easier to be implemented.

However, the partway deadheaded strategy needs some additional cost (i.e., the deadheaded bus runs empty to a stop of the peak direction and restart its service). To avoid invalid partway deadheading operation, the reliability of the further transit service was firstly assessed based on the current and recent service reliabilities. Only when further service was unreliable, the partway deadheading operation should be implemented. For determining the beginning stop for the new service of the deadheaded bus, an optimization model aiming to maximize the benefit of transit system was presented. To address service reliability assessment and partway deadheading operation optimization, a heuristic algorithm has been defined and implemented.

Then, the partway deadheading strategy proposed in this study was tested with the data from a real-life route in Dalian city of China. The length of rolling horizon and the threshold value of service reliability were analyzed. Furthermore, the performance of the partway deadheading strategy was validated and the results showed that the proposed strategy can yield the best

performance compared to other partway deadheading strategies and no control. In addition, the heuristic algorithm was suitable for online application due to its simplicity in computation, whose running time on a PC was no more than a few seconds.

Acknowledgments

This work was supported in National Natural Science Foundation of China 51108053, Humanities and Social Sciences Foundation from the Ministry of Education of China 10YJC630357 and the Fundamental Research Funds for the Central Universities 2011ZC029 and 2011QN037.

References

Bay, S., 1999. Nearest neighbor classification from multiple feature subsets. Intelligent Data Analysis 3 (3), 191–209.
Casello, J.M., Nour, A., Hellinga, B., 2009. Quantifying impacts of transit reliability on user costs. Transportation Research Record 2112 (145), 136–141.
Ceder, A., Stern, H.I., 1981. Deficit function bus scheduling with deadheading trip insertions for fleet size reduction. Transportation Science 15 (4), 338–363.
Chen, X., Yu, L., Zhang, Y., Guo, J., 2009. Analyzing urban bus service reliability at the stop, route, and network levels. Transportation Research Part A 43 (8), 722–734.
Cheng, C.T., Xie, J.X., Chau, K.W., Layeghifard, M., 2008. A new indirect multi-step-ahead prediction model for a long-term hydrologic prediction. Journal of Hydrology 361 (1–2), 118–130.
Dessouky, M., Hall, R., Zhang, L., Singh, A., 2003. Real-time control of buses for schedule coordination at a terminal. Transportation Research Part A 37 (2), 145–164.
Delle Site, P., Filippi, F., 1998. Service optimization for bus corridors with short-turn strategies and variable vehicle size. Transportation Research 32A, 19–38.
Eberlein, X.J., Wilson, N.H.M., Barnhart, C., Bernstein, D., 1998. The real-time deadheading problem in transit operations control. Transportation Research Part B 32 (2), 77–100.
Eberlein, X.J., Wilson, N.H.M., Bernstein, B., 2001. The holding problem with real-time information available. Transportation Science 35 (1), 1–18.
Eberlein, X., Wilson, N.H.M., Bernstein, D., 1999. Modeling real-time control strategies in public transport operations. In: Computer-aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems, vol. 471. Springer-Verlag, pp. 352–346.
Fu, L., Liu, Q., Calamai, P., 2003. Real-time optimization model for dynamic scheduling of transit operations. Transportation Research Record 1897, 48–55.
Furth, P.G., 1985. Alternating deadheading in bus route operations. Transportation Science 19 (1), 13–28.
Furth, P.G., 1987. Short turning on transit routes. Transportation Research Record 1108, 42–52.
Kimpel, T.J., Strathman, J., Bertini, R.L., Callas, S., 2005. Freeway operations, high-occupancy vehicle systems. Transportation Research Record 1925, 156–166.
Lin, J., Wang, P., Barnum, D.T., 2008. A quality control framework for bus schedule reliability. Transportation Research Part E 44 (6), 1086–1098.
Miller, E.J., Bunt, P.D., 1987. Simulation model of shared right-of-way streetcar operations. Transportation Research Record 1152, 31–41.
Polus, A., 1978. Modeling and measurement of bus service reliability. Transportation Research 12, 253–256.
Shen, S., Wilson, N.H.M., 2001. An optimal integrated real-time disruption control model for rail transit systems. In: Voss, S., Daduna, J.R. (Eds.), Computer-aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems, vol. 505. Springer-Verlag, pp. 335–364.
Silcock, D.T., 1981. Measures of operational performance for urban bus services. Traffic Engineering and Control 22, 645–648.
Smith, B.L., Williams, B.M., Oswald, R.K., 2002. Comparison of parametric and nonparametric models for traffic flow forecasting. Transportation Research Part C 10 (4), 303–321.
Sorratini, J.A.P., Liu, R., Sinha, S., 2008. Assessing bus transport reliability using micro-simulation. Transportation Planning and Technology 31 (3), 303–324.
Strathman, J.G., Hopper, J.R., 1993. Empirical-analysis of bus transit on-time performance. Transportation Research Part A – Policy and Practice 27 (2), 93–100.
Strathman, J.G., Kimpel, T.J., Ducker, K.J., Gerhart, R.L., Turner, K., Griffin, D., Callas, S., 2001. Bus transit operations control: review and an experiment involving TRIMET's automated bus dispatching system. Journal of Public Transportation 4, 1–26.
van Oort, N., van Nes, R., 2009. Regularity analysis for optimizing urban transit network design. Public Transport 1 (2), 155–168.
van Oort, N., Wilson, N.H.M., van Nes, R., 2010. Reliability Improvement in Short Headway Transit Services: Schedule-Based and Headway-Based Holding Strategies. The 89th Annual Meeting of the Transportation Research Board, Hilton, Cabinet.
Vijayaraghavan, T.A.S., Anantharamaiah, K.M., 1995. Fleet assignment strategies in urban transportation using express and partial services. Transportation Research 29A, 157–171.
Yang, Z.Z., Yu, B., Cheng, C.T., 2007. A parallel ant colony algorithm for bus network optimization. Computer-Aided Civil and Infrastructure Engineering 22 (1), 44–55.
Yu, Bin, Lam, William, H.K., Tam, Mei Lam, 2011. Bus arrival time prediction at bus stop with multiple routes. Transportation Research Part C 19 (6), 1157–1170.
Yu, Bin, Yang, Zhongzhen, 2009. A dynamic holding strategy in public transit systems with real-time information. Applied Intelligence 31 (1), 69–80.
Yu, B., Yang, Z.Z., Jin, P-H., Wu, S.H., Yao, B.Z., 2012. Transit route network design-maximizing direct and transfer demand density. Transportation Research Part C 22, 58–75.
Zhao, J., Dessouky, M., Bukkapatnam, S., 2006. Optimal slack time for schedule-based transit operations. Transportation Science 40 (4), 529–539.

Transportation Research Part C 19 (2011) 1157–1170

Contents lists available at ScienceDirect

Transportation Research Part C

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

Bus arrival time prediction at bus stop with multiple routes

Bin Yu [a,b,*], William H.K. Lam [a], Mei Lam Tam [a]

[a] Department of Civil and Structural Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, PR China
[b] Transportation Management College, Dalian Maritime University, Dalian, PR China

ARTICLE INFO

Article history:
Received 1 September 2010
Received in revised form 19 January 2011
Accepted 21 January 2011

Keywords:
Bus arrival time prediction
Multiple bus routes
Support vector machine
Artificial neural network
k nearest neighbours algorithm

ABSTRACT

Provision of accurate bus arrival information is vital to passengers for reducing their anxieties and waiting times at bus stop. This paper proposes models to predict bus arrival times at the same bus stop but with different routes. In the proposed models, bus running times of multiple routes are used for predicting the bus arrival time of each of these bus routes. Several methods, which include support vector machine (SVM), artificial neural network (ANN), *k* nearest neighbours algorithm (*k*-NN) and linear regression (LR), are adopted for the bus arrival time prediction. Observation surveys are conducted to collect bus running and arrival time data for validation of the proposed models. The results show that the proposed models are more accurate than the models based on the bus running times of single route. Moreover, it is found that the SVM model performs the best among the four proposed models for predicting the bus arrival times at bus stop with multiple routes.

1. Introduction

1.1. Backgrounds

Many applications of information systems and technologies, such as automatic vehicle location (AVL) or identification (AVI) systems and automatic passenger counters (APC), are now receiving increasing attentions in transportation management. They are considered as the key components of intelligent transportation systems (ITS). Recently, transit agencies also realize the operational benefits of ITS-related technology implementation. Based on these advanced technologies, transit agencies can acquire real-time bus information to reduce passenger journey time and improve management/service level. Thus, there is a growing interest in providing real-time bus arrival information for passengers using emerging electronic information and communication technologies. The availability of real-time bus information can help passengers efficiently schedule their departure time and make smart choices for their travel.

In practice particularly in transit-oriented cities like Hong Kong, it is very common to have several bus routes using the same bus stop in urban areas. In the bus stop with multiple routes, passengers would have several choices (different bus routes) to reach their destinations. Real-time bus information available at stop can be very helpful to passengers if they can know which bus will arrive first. Thus, there is a need for bus information at stop with multiple routes particularly in high density populated cities (e.g., Hong Kong) with large bus passenger demands. For example, there are nineteen bus routes using two bus stops (i.e. two separate bus bays and some stop boards at each bus bay) at the Cross Harbour Tunnel (North bound) in Hong Kong. Among these bus routes, there are some common lines passing several major urban areas on

* Corresponding author at: Department of Civil and Structural Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, PR China.
E-mail address: minlfish@yahoo.com.cn (B. Yu).

0968-090X/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trc.2011.01.003

Hong Kong Island. There is always a long queue of passengers waiting for buses at the bus stop board of each route. Therefore, given the bus arrival time information at the bus stop with multiple routes, passengers who have multiple choices can choose the potentially suitable route for their journeys. With this real-time bus arrival time information, passengers can greatly reduce their waiting times and the efficiency of bus services can significantly be improved particularly at the major bus stops with high volume of passengers transferring from railway to bus modes.

Fig. 1 shows an example for illustrating the effort on providing bus arrival time information at bus stop with multiple routes. In the example, a passenger expects to go from Stop A to Stop B where there are three bus routes, route nos. 101, 102 and 103. The passenger can choose any one of these three bus routes to his destination at Stop B. If the passenger knows the bus arrival times of the next buses of the three bus routes at Stop A (e.g., 09:05, 09:04 and 09:02, respectively), he/she will wait for the next bus of the route no. 103 rather than the other two routes. Thus, his/her waiting time will be reduced.

Passengers, in general, are more interested in knowing the predicted arrival times of the following buses rather than the current locations of the buses. Thus, the accuracy of the prediction algorithm is very important in a successful bus information system. However, accurate prediction of bus arrival time is very difficult due to many stochastic variables involved (e.g., traffic conditions). Therefore, the deployment of bus arrival time prediction model is a challenging task.

1.2. Literature review

In the past decade, various sophisticated techniques and algorithms have been developed to predict bus travel time or arrival time by using AVL and/or APC data. These methods can be categorized as: artificial neural network (ANN) or Support vector machine (SVM) (Ding and Chien, 2000; Chien et al., 2002; Chen et al., 2004; Jeong and Rilett, 2004; van Lint et al., 2005; Vlahogianni et al., 2005; van Hinsbergen et al., 2009; Yu et al., 2006, 2010a,b), Non-parametric regression (NPR) model (Smith et al., 2002; Chang et al., 2010; Park et al., 2007; Zhang and Rice, 2003; You and Kim, 2000; Vlahogianni et al., 2006) and Kalman filter model (Wall and Dailey, 1999; Chien and Kuchipudi, 2003; Shalaby and Farhan, 2004; Chen et al., 2004; Yu et al., 2010a).

1.2.1. Artificial neural network/support vector machine models

ANN is motivated by emulating the intelligent data processing ability of human brains. ANN has been reported to be especially useful for finding solutions for complex non-linear problems. Chien et al. (2002) proposed two ANN-based models: the link-based ANN model and the stop-based ANN model, to predict bus arrival time. An adaptive algorithm was also developed to improve the performances of the ANN-based models. Their results showed that the link-based ANN model outperformed the stop-based ANN model for the prediction with a relatively small number of intersections. The results also indicated that the adaptive algorithm can improve the performances of the ANN-based models. Chen et al. (2004) proposed a dynamic algorithm that integrated the ANN model and a Kalman filter-based algorithm. Jeong and Rilett (2004) compared the performances of several methods: the historical data based model, the regression models, and the ANN model, for bus arrival time prediction. Their results showed that the ANN model outperformed the historical data based model and the regression model in terms of prediction accuracy. Van Lint et al. (2005) presented a freeway travel time prediction framework which combined state-space neural network with preprocessing strategies based on imputation. Their results indicated that a combination of these imputation procedures and the proposed model could be implemented a real-time application. Vlahogianni et al. (2005) presented a multilayered structural optimization strategy based on genetic algorithm, which was applied to both univariate and multivariate traffic flow data to evaluate the performance of the developed network. van Hinsbergen et al. (2009) used Bayesian inference theory to combine neural networks in a committee using. The proposed method had an evidence factor to act as a criterion of stopping the training and a tool to determine different neural networks. Their results showed that the proposed approach had a much higher accuracy.

SVM is a very specific type of learning algorithm characterized by the capacity control of the decision function, the use of the kernel functions, and the sparse solution (Cristianini and Shawe-Taylor, 2000; Vapnik, 1999, 2000). Yu et al. (2006, 2010b) developed the SVM-based models to predict bus arrival time. In the models, travel speeds of the preceding buses of the same bus route were used to estimate traffic conditions. Their results showed that the SVM-based models outperformed the ANN and historic mean prediction models, and SVM seemed to be a powerful alternative for bus arrival time prediction. With versatile parallel distributed structures and adaptive learning processes, ANN and SVM have recently been gaining popularity in bus travel/arrival time prediction.

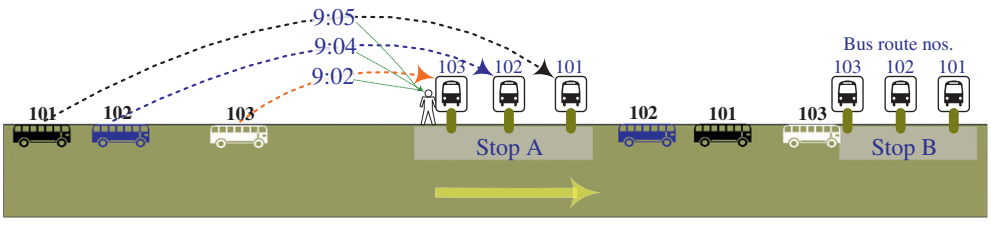


Fig. 1. An example for bus arrival time information at bus stop with multiple routes.

1.2.2. Non-parametric regression models

NPR is a relatively simple method for prediction without the need to estimate parameters. The NPR models are therefore more suitable for real-time applications. Zhang and Rice (2003) proposed a method to predict short-term freeway travel times using a linear model in which the coefficients varied as smooth functions of the departure time changed. Several varieties of the prediction procedures were presented and the results were encouraging. You and Kim (2000) developed a hybrid travel time forecasting model based on non-parametric regression for predicting link travel times in congested road networks.

Recently, k nearest neighbour (k-NN) is one of the most popular NPR methods, which has been widely applied in many fields (Smith et al., 2002; Chan et al., 2009; Tam and Lam, 2009). Chang et al. (2010) developed a bus travel time prediction model using k-NN. Their results showed that k-NN was an effective method to estimate bus travel time according to the prediction accuracy and computing time. Park et al. (2007) applied a non-parametric regression model for travel time prediction. Their model was implemented and evaluated using real-time transit data. k-NN methods have also been documented while a large database is desirable for increasing the prediction accuracy. However, the large sample size has significant implications on the timeliness of model execution (Smith et al., 2002).

1.2.3. Kalman filter models

Kalman filter is an efficient recursive procedure that estimates the future states of dependent variables. It is originated from the state-space representations in modern control theory. Chien and Kuchipudi (2003) developed a path-based model and a link-based model using Kalman filter to predict bus travel times. Their results showed that the link-based model would be more sensitive to travel time increment of the link with congestion or incident. Shalaby and Farhan (2004) discovered that the Kalman filter-based model outperformed the regression and neural network models in terms of accuracy. Cathey and Dailey (2003) proposed a prediction method for bus arrival/departure time which included three components, e.g., a track, a filter and a predictor. In the filter, Kalman filter was used to estimate the vehicle dynamical state. Chen et al. (2004) developed an enhanced algorithm based on Kalman filter to predict bus arrival time. Their results showed that the enhanced algorithm was effective for bus arrival time prediction compared with the standard ANN-based models.

In summary, many researches have been conducted on forecasting bus travel/arrival time for a single bus route, while few studies have investigated bus travel/arrival time prediction for multiple bus routes. In most previous researches, some factors related to road traffic (e.g., traffic speed and volume) were used to model bus travel/arrival time prediction. However, bus running is greatly different from that of other vehicles due to bus lane, bus stop and so on. To solve the problems, Yu et al. (2006, 2010b) applied the bus running times (speeds) of the preceding buses to predict the arrival time of the next bus. However, in their researches, the bus running times (speeds) of the same route were used to model the prediction. For the route segment passing through several bus routes, the bus arrival time prediction model can provide better prediction accuracy by integration of bus information of multiple routes.

1.3. Contributions

In general, bus operation is different for the buses arriving at stops with multiple routes or single route. Bus arrival time at stops with single route is mainly affected by traffic conditions between stops. However, besides traffic condition, bus arrival time at stops with multiple routes would also be influenced by buses of other bus routes. For example, limited capacity of bus stop may cause buses queue up at the bus stop. As a result, bus arrival delays would be increased. This is very common in urban areas of transit-oriented cities like Hong Kong, particularly during peak periods.

As to the predictions for single route or multiple routes, there is some difference in the development of model. Fig. 2 shows the difference between the predictions for single route and multiple routes. When the target bus n of the bus route no 101 arrives at Location A, assume the buses $k,\ldots, k+\mu,\ldots, k+\delta$, have gone through Bus Stop, i.e., the buses are the preceding buses. Since the running times of the preceding buses between Location A and Bus Stop can be obtained, the running times are used to estimate traffic condition and to construct the prediction model. For the prediction for single route, only the running time of bus (e.g., the bus $k+\mu$) on the same route is used to model the prediction. Thus, the performance of the

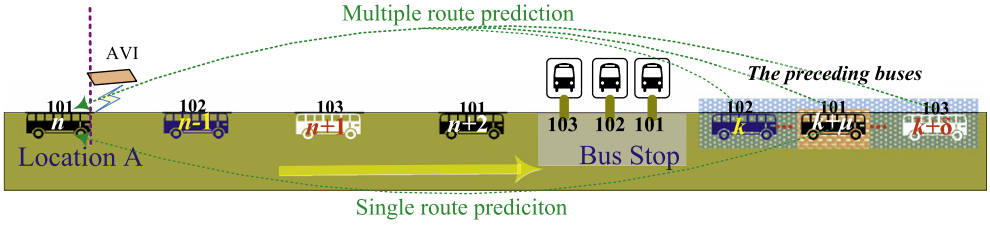


Fig. 2. The difference between the predictions of single route and multiple routes.

single route prediction model greatly relies on the timeliness and accuracy of the information of the preceding buses on the same route. However, except for buses on the same route, the bus information from other routes is also used for the multiple route prediction model. Compared with the bus information of the same route, multi-dimension bus information from multiple routes can provide more benefit (e.g., timeliness and reliability of the information), which will certainly improve prediction accuracy.

This paper seeks to make two contributions to the literature. Firstly, it attempts to develop the models to predict bus arrival time at the same bus stop but with multiple routes using real-world data. It is expected that the anxieties and waiting times of passengers can be reduced if passengers know when the next buses of multiple routes arrive at the bus stop. Bus running times of different routes are considered in the proposed models for predicting the bus arrival time. Secondly, the performances of several prediction methods, SVM, ANN, *k*-NN and linear regression (LR), are assessed and compared for forecasting bus running time with multiple routes. The performance comparison of several models can provide valuable insight for researchers as well as practitioners.

The structure of this paper is organized as follows: Section 2 provides the formulation of four proposed models for predicting bus arrival time at bus stop with multiple routes; Section 3 presents a case study together with results and analysis including performance evaluation of the four proposed models; and lastly, the conclusions are given in Section 4 together with suggestions for further study.

2. Methodologies

2.1. Prediction framework

Bus arrival time prediction at bus stop with multiple routes can be described in the following way: Given the bus arrival time of any bus route at a location, it is to predict the bus arrival time at the bus stop with multiple routes. Fig. 3 illustrates an example for the framework of the prediction in this study. When a bus (n) of any bus route (l) arrives at the Location A, the bus arrival time ($T^a_{l,n}$) can be recorded by some traffic data collection technologies (such as AVI). Then, the running time ($\hat{t}^{running}_{l,n}$) between the stop and the Location A is predicted by some methods. According to the arrival time of the bus at the Location A, the bus arrival time ($\hat{T}^s_{l,n}$) at the stop with multiple routes can be determined.

$$\hat{T}^s_{l,n} = T^a_{l,n} + \hat{t}^{running}_{l,n} \tag{1}$$

To predict bus running time in an accurate and timely manner, it is essential to determine the appropriate factors to estimate traffic conditions. Yu et al. (2006, 2010b) suggested that the running time(s) of the preceding bus(es) that has(ve) just reached the stop can be used to reflect the traffic conditions. Furthermore, they also pointed out the weighted average running times of several preceding buses could reduce the effect of accidents on the preceding buses. However, in the researches by Yu et al. (2006, 2010b), the bus running times of only the same bus route were applied to estimate traffic conditions. In fact, if integrating bus running times of different bus routes, the estimation accuracy of traffic conditions can be improved. In general, the most up-to-minute data can provide most reliable information for the prediction. Therefore, in order to take into account the timeliness of the bus running times of the preceding buses, the time headway between the target bus and the last preceding bus that has just reached the stop is considered in the proposed models.

For the sake of simplicity, "the preceding bus(es)" denotes the last bus(es) that has(ve) just reached the stop in the following sections. Assuming that n represents the target bus at the Location A, l represents the route no. of the bus n and L represents the set of bus routes. The factors considered in this study can be illustrated as follows.

(a) $t^l_{L,n}$ is the time headway between the target bus and the last preceding bus of any route among the route set L. The last preceding bus may be the same bus route or different bus route with the target bus. Thus, $t^l_{L,n}$ can be obtained by the following equation.

$$t^l_{L,n} = T^a_{l,n} - T^a_{L,k} \tag{2}$$

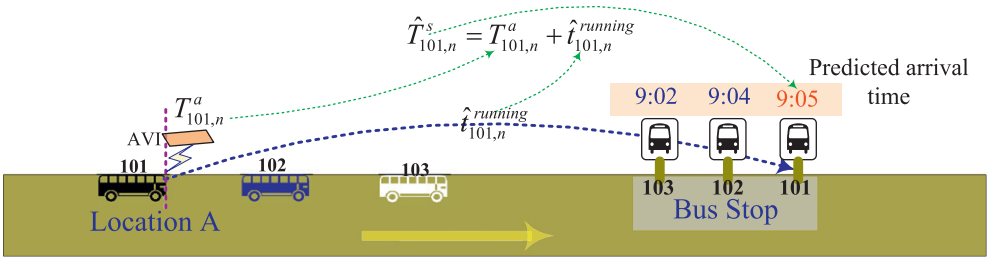


Fig. 3. An example for the prediction framework.

where $T^a_{l,n}$ represents the arrival time of the bus (n) of the bus route no. l at Location A. The bus k represents the last preceding bus of any route among the route set L. $T^a_{L,k}$ represents the arrival time of the bus (k) at Location A.

(b) $t^i_{l,n}$ is the time headway between the target bus and the last preceding bus of the same route no. l. Fig. 4 illustrates the difference between the two variables, $t^l_{L,n}$ and $t^i_{l,n}$.

$$t^i_{l,n} = T^a_{l,n} - T^a_{l,k+\mu} \tag{3}$$

where the bus $k+\mu$ represents the last preceding bus of the same bus route (l).

(c) $\bar{t}^r_{L,n}$ is the weighted average running time of several preceding buses (e.g., the bus k,..., $k+\mu$,..., $k+\delta$) of any routes among the route set L. In general, the last preceding bus to the target bus will contribute more to the weighted average running time than the further ones. A simple weighted method is to give each preceding bus a weight of the inverse of the time headway between the preceding bus and the target bus. Thus, $\bar{t}^r_{L,n}$ can be attained by the following equations.

$$\bar{t}^r_{L,n} = \sum_{j=1}^{\delta} \frac{1/(T^a_{l,n} - T^a_{L,j})}{\Gamma}(t^r_{L,j}) \tag{4}$$

$$\Gamma = \sum_{j=1}^{\delta} 1/(T^a_{l,n} - T^a_{L,j}) \tag{5}$$

where, $t^r_{L,j}$ is the running time between Location A and the stop of the jth preceding bus. Γ is the sum of the weight of each preceding bus. δ is the prediction horizon that is the number of the selected preceding buses.

(d) $t^r_{l,n}$ is the running time of the preceding bus (e.g., the bus $k+\mu$ in Fig. 3) of the same bus route No. l.

$$t^r_{l,n} = T^s_{l,k+\mu} - T^a_{l,k+\mu} \tag{6}$$

where $T^s_{l,k+\mu}$ is the arrival time of the bus ($k+\mu$) of the route No. l at the stop.

If $\hat{t}^{running}_{l,n}$ represents the prediction of bus running time between Location A and the stop, the prediction model for bus running times aims to generalize the relationship of the following form.

$$\hat{t}^{running}_{l,n} = f(t^l_{L,n}, t^i_{l,n}, \bar{t}^r_{L,n}, t^r_{l,n}) \tag{7}$$

To develop the bus running time prediction model, several techniques are employed in this study. With versatile parallel distributed structures and adaptive learning processes, ANN and SVM appear to be the suitable approaches for bus running prediction (Ding and Chien, 2000; Chien et al., 2002; Yu et al., 2006, 2010b). Thus, ANN and SVM are used to model bus running time prediction in this study. In addition, as simple and effective regression techniques, both k-NN and LR are also used for comparison.

2.2. Support vector machine models

SVM is a type of learning algorithms based on statistical learning theory, which can be adjusted to map the input–output relationship for the non-linear system. In addition, the solution of SVM is always unique and globally optimal since training SVM is equivalent to solving a linearly constrained quadratic programming problem. Therefore, SVM shows the strong resistance to the over-fitting problem and the high generalization performance. It is mainly because SVM can construct a mapping from one-dimensional input vector into high-dimensional space by the use of reproducing kernels. Here, Fig. 5 shows that the SVM-based model for the prediction of the bus arrival time at the stop with multiple routes.

2.3. Artificial neural network

ANN is a mathematical model by simulating the neural structure of the human brain. The ANN processes information by means of interaction between many neurons and the different links between neurons have been associated with weights. Based on the highly interconnected neural computing elements, ANN has the ability to model complex relationships

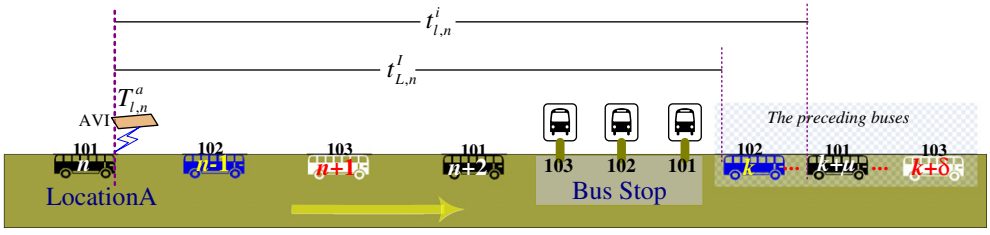


Fig. 4. The difference between variables $t^l_{L,n}$ and $t^i_{l,n}$.

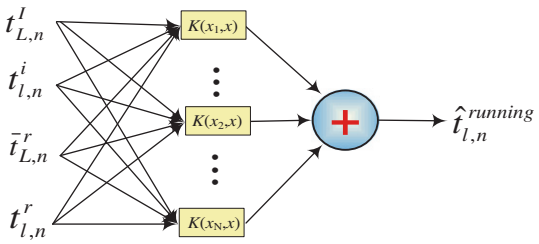


Fig. 5. Structure of the SVM model for the bus arrival time at the stop with multiple routes.

between inputs and outputs to find patterns in data. ANN includes two working phases, the learning phase and the recalling phase. During the learning phase, learning means using a set of observations are commonly used as a training signal in input and output layers. The recalling phase is performed by one pass using the weight obtained in the learning phase. ANN with three layers is chosen in this study as it is generally easy to use and can approximate almost any input/output relationships. The fully connected multilayer feed forward neural network with a back propagation (BP) algorithm has been applied successfully to deal with complex transportation systems (Huang and Ran, 2003). Hence, the neural network used to predict bus running time described as Fig. 6 in this study.

2.4. k nearest neighbours algorithm

k-NN is a method for classifying objects based on closest observations in a feature space. *k*-NN method is one of the simplest machine learning algorithms. In *k*-NN method, the Euclidean distance is usually used to determine the distance between the input state and the historical data in the feature space. On the basis of the Euclidean distance, the *k* nearest neighbours with the least distance to the input state can be determined. To weight the contributions of each neighbour, a common distance-based scheme is adopted to compute the weight of each neighbour. The forecasts can then be obtained by taking the weighted average of the observations from the *k* nearest neighbours.

In traditional *k*-NN method, standard Euclidean distance is used to match the *k* nearest neighbours in the feature space. This means that each independent variable has the same importance to the input state. For the prediction of bus running time, the equal weight of the independent variables is unreasonable. In this study, a weighed distance (d_j) is introduced to assign higher weight to the more important independent variable.

$$\hat{t}^{running}_{l,n} = \sum_{j=1}^{k} \frac{1/d_j}{D}(t^r_{L,j}) \tag{8}$$

$$d_j = \sqrt{\frac{\lambda_1 \times (t^I_{L,n} - t^I_{L,k})^2 + \lambda_2 \times (t^i_{l,n} - t^i_{l,k})^2 + \lambda_3 \times (\bar{t}^r_{L,n} - \bar{t}^r_{L,k})^2 + \lambda_1 \times (t^r_{l,n} - t^r_{l,k})^2}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4}} \tag{9}$$

$$D = \sum_{j=1}^{k} \frac{1}{d_j} \tag{10}$$

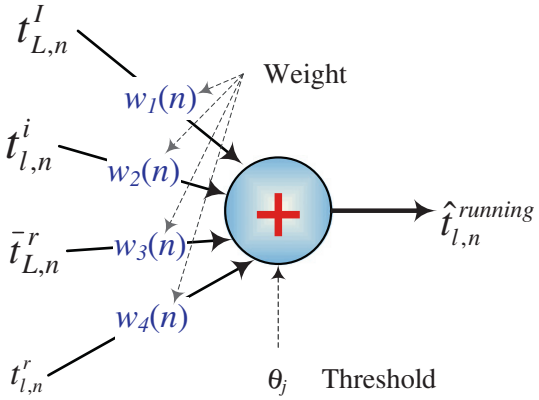


Fig. 6. Structure of the ANN model for the bus arrival time at the stop with multiple routes.

where d_j represents the weighted distance between the jth nearest neighbour and the input state. D represents the sum of the weighted distance of the k nearest neighbours. λ_1, λ_2, λ_3 and λ_4 represent the weights of the variables. The values of λ_1, λ_2, λ_3 and λ_4 are equal to the correlation coefficients between each independent variable and the dependent variable in this study.

2.5. Linear regression

Linear regression (LR) is the first type of regression analysis and used extensively in practical applications. For bus running time prediction, linear regression method is to model the relationship between the estimated bus running time (dependent variable) and the impact factors (independent variables).

Because the relationship between the estimated bus running time and the information of the preceding buses is very sophisticated, linear regression is extended with some interaction to make LR a more valid comparison with other models in this study. Here, the logarithm of the data set $\{\hat{t}_{l,n}^{running} : t_{L,n}^{l}, t_{l,n}^{i}, \bar{t}_{L,n}^{r}, t_{l,n}^{r}\}$ is taken, and then a new data set $\{\ln(\hat{t}_{l,n}^{running}) : \ln(t_{L,n}^{l}), \ln(t_{l,n}^{i}), \ln(\bar{t}_{L,n}^{r}), \ln(t_{l,n}^{r})\}$ is obtained. For the bus arrival time of the stop with multiple routes, linear regression model assumes that the relationship between the dependent variable and the independent variables (after the logarithm transformation) is approximately linear. Then, the approximate relationship is modeled as follow:

$$\ln(\hat{t}_{l,n}^{running}) = \beta_1 \times \ln(t_{L,n}^{l}) + \beta_2 \times \ln(t_{l,n}^{i}) + \beta_3 \times \ln(\bar{t}_{L,n}^{r}) + \beta_4 \times \ln(t_{l,n}^{r}) \tag{11}$$

where, β_1, β_2, β_3 and β_4 are coefficients that are related to the effects of impact factors on bus running time.

3. Case study

In this section, the proposed several models for predicting bus arrival time at bus stop with multiple routes have been evaluated by the real-world data in Hong Kong. Hong Kong has a highly developed and sophisticated bus route network that comprises about 700 bus routes. Over 90% of the daily journeys are on public transport, making it the highest rate in the world. In Hong Kong, real-time travel information system (RTIS) provides area-wide traffic information in the whole network (Tam and Lam, 2008). In RTIS, real-time traffic data (Autotoll tag records) are collected by AVI technology. The Autotoll tag records are initially used for electronic toll collection in Hong Kong. Almost all the buses have been installed with Autotoll tags for toll collection automatically.

Bus stop near the entrance of the Cross Harbour Tunnel (CHT) (North bound) in Kowloon Central urban area is selected for testing the proposed models. This bus stop is chosen because there are many bus routes with large passenger demands on harbour crossing every day. According to the locations of Autotoll tag readers, two directions, the west direction from Chatham Road North (CRN) to the CHT and the east direction from Ping Chi Street (PCS) to the CHT, are chosen. The locations of the bus stop near the CHT and the Autotoll tag readers are illustrated in Fig. 7, respectively. There are eight bus routes, operating along the west direction (through CRN), which include route nos. 102, 103, 104, 110, 112, 117, 118 and 171. Bus routes, operating along the east direction (through PCS), include route nos. 101, 107, 108, 109, 111 and 116. The distances from CRN and PCS to the CHT bus stop are about 0.62 km and 0.72 km, respectively.

3.1. Performance measures

The prediction results are evaluated in terms of the performances of three measures; namely, the mean absolute error (MAE_l), the mean absolute percentage error ($MAPE_l$) and the root mean square error ($RMSE_l$) of the route no. l. The three terms can judge the difference between the observed and the predicted running time in different aspects.

$$MAE_l = \frac{\sum |t_{l,n}^{running} - \hat{t}_{l,n}^{running}|}{N} \tag{12}$$

$$MAPE_l = \frac{1}{N} \sum \frac{|t_{l,n}^{running} - \hat{t}_{l,n}^{running}|}{t_{l,n}^{running}} \times 100\% \tag{13}$$

$$RMSE_l = \sqrt{\frac{\sum (t_{l,n}^{running} - \hat{t}_{l,n}^{running})^2}{N-1}} \tag{14}$$

where $t_{l,n}^{running}$ is the observed running time of the bus n of the route no. l. $\hat{t}_{l,n}^{running}$ is the predicted running time of the bus n of the route no. l. N is the number of the buses which have been observed.

3.2. Data collection and processing

To obtain the actual bus running and arrival time data, the video surveys near the CHT bus stop have been carried out in typical weekdays on 11–12 May 2010 (Tuesday to Wednesday) and 8 June 2010 (Tuesday) during the morning peak

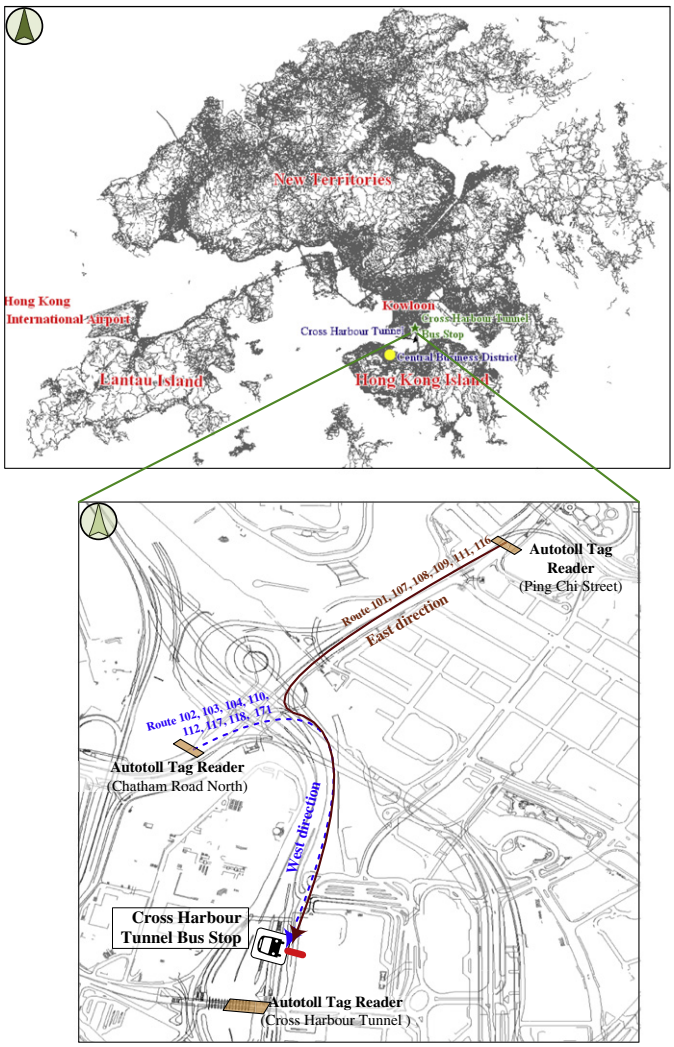


Fig. 7. Locations of the survey paths and the Autotoll tag readers.

(08:00–10:00). In the surveys, we have recorded the route no. and license plate of each bus passing through the CHT bus stop. Then, by a manual license plate matching with the Autotoll tags records, the actual bus arrival time at the CHT bus stop and the time passing through the Autotoll tag readers (CRN and PCS) can be acquired. The data recorded by the Autotoll tag readers at the toll-booths of the CHT have been used to check the license plate reading and the time.

A data filtering algorithm (Tam and Lam, 2008) was then applied to the observations collected from the surveys in order to filter out the outliers. The numbers of the valid observations on the 3 days are 237, 228 and 224, respectively. We divide these routes into two groups according to the different directions (CRN and PCS to CHT stop). Table 1 shows the number of the valid observations of each route at each day, and the main descriptive statistics for the collected travel time. From Table 1, the average bus travel time of the west direction (from CRN to the CHT bus stop) is obviously more than that of the east direction (from PCS to the CHT bus stop). The bus running times of the east direction varied from 170 to 485s and the average time is around 291s. The bus running times of the west direction are from 275 to 662s and the average time is around 449s. RMSEs of the east and west directions are 57.5 and 67.3s, respectively.

3.3. Model identifications

Before model identifications, the parameter δ of the weighted average running time should be determined. By sensitivity tests, bus running times of three preceding buses are used to calculate the weighted average running time in this study, i.e., $\delta = 3$. In model identifications, the observations are first classified by the bus route no. and the inputs of the prediction models are computed. Then, the observations on 11 May 2010 are set aside as testing data. The observations on 12 May and 8 June 2010 are selected as training data to calibrate the prediction models. To have the same basis of comparison, the same training and verification sets are used for all models.

Table 1
Sample sizes of each route and descriptive statistics for the collected data.

Route no.		Sample sizes (vehicle)			Descriptive statistics			
		11-May	12-May	8-June	Min (s)	Max (s)	Ave (s)	RMSE
PCS–CHT bus stop	101	32	29	26	197	424	300.40	61.29
	107	13	12	16	171	485	307.63	68.34
	108	11	9	8	229	436	292.72	45.43
	109	10	9	10	193	409	267.62	56.61
	111	29	30	34	170	456	291.58	58.46
	116	23	22	21	176	432	287.90	55.15
CRN–CHT bus stop	102	20	17	15	327	662	455.14	77.82
	103	11	8	8	360	584	442.62	60.22
	104	27	30	29	315	621	432.40	61.01
	110	7	8	6	369	616	472.39	72.52
	112	14	16	12	302	629	448.73	70.85
	117	5	6	7	352	562	442.06	63.98
	118	18	14	17	275	637	457.60	76.94
	171	17	18	15	308	639	438.43	66.89

Table 2
SVM and ANN models with different input parameters.

Model	Input parameters				Average MAE (s)	
	Bus time interval among the route set ($t^l_{L,n}$)	Bus time interval of the same route ($t^i_{l,n}$)	Weighted average bus running time among the route set ($\bar{t}^r_{L,n}$)	Bus running time of the same route ($t^r_{l,n}$)	SVM	ANN
1	√	√	√		32.82	**34.34**
2		√	√	√	33.37	38.69
3	√	√		√	34.34	36.34
4	√		√	√	35.21	37.01
5	√	√	√	√	**30.39**	35.02
6		√		√	37.76	42.22

3.3.1. Support vector machine models

The previous researches (Yu et al., 2006, 2010b) suggested that radial basis function (RBF) kernel was efficient for bus running time prediction. Thus, RBF kernel function is used for the SVM model in this study. To determine the inputs of the SVM model, sensitivity tests have been conducted. The SVM models with different input variables listed in Table 2 have been calibrated based on the data collected. Table 2 also showed that the average MAE of the prediction for all the routes using the SVM models with different parameters. The prediction errors of each route are shown in Fig. 8.

The models 1–5 integrate bus information of multiple routes to predict the bus arrival time at the bus stop. The model 6 is a standard method using bus information of a single route for the prediction. Obviously, the performance of the single route prediction model is the worst among the six models. This indicates that integrating bus information of multiple routes can improve the accuracy of the arrival time prediction at the bus stop with multiple routes. It is mainly because that bus information of multiple routes can reduce the effect of accidents on the preceding buses that may be the same or different route with the target bus. Furthermore, Fig. 8 also shows that the SVM model 5 can almost provide the best prediction accuracy for each route. Thus, all the four variables $t^l_{L,n}, t^i_{l,n}, \bar{t}^r_{L,n}, t^r_{l,n}$ are considered as the inputs of the SVM model in this study.

Before applying SVM, there are two parameters, C and ε, which are first determined. Parameter C is to determine the trade-off between the model complexity and the degree in the optimization equation. Parameter ε controls the width of the ε-insensitive zone which is used to fit the training data. Referring to the application of bus running time prediction from Yu et al. (2006, 2010b), it is recommended to the constraints of the two parameters which respectively attribute to the range $C \in [2^{-5}, 2^5]$ and $\varepsilon \in [0.1, 0.3]$.

As identifying the parameters in SVM, grid-search is used to pick up the optimal parameter values. Thus, for the bus running time prediction, the two parameters (C, ε) are selected as (2, 0.1).

3.3.2. Artificial neural network

A standard three-layer ANN is used to construct the prediction model for bus running time in this study. Similar to the SVM model, the input parameters of the ANN model are determined based on the results of the sensitivity tests. The combinations of input parameters of the ANN models are the same as the ones of the sensitivity tests of the SVM model. The ANN

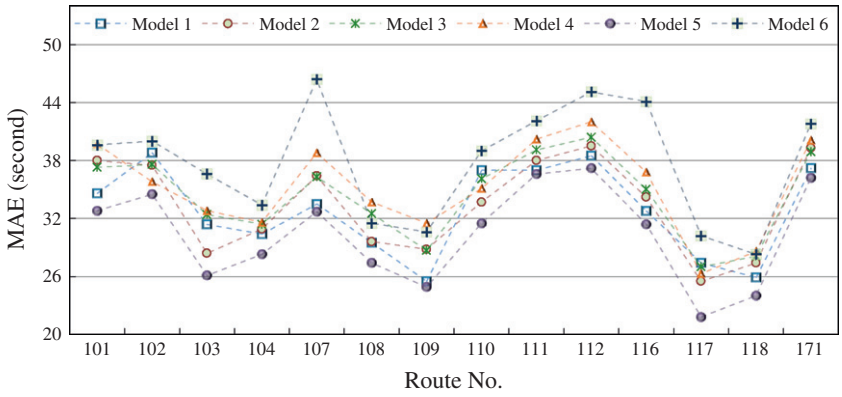


Fig. 8. Prediction errors of the SVM models with different input parameters.

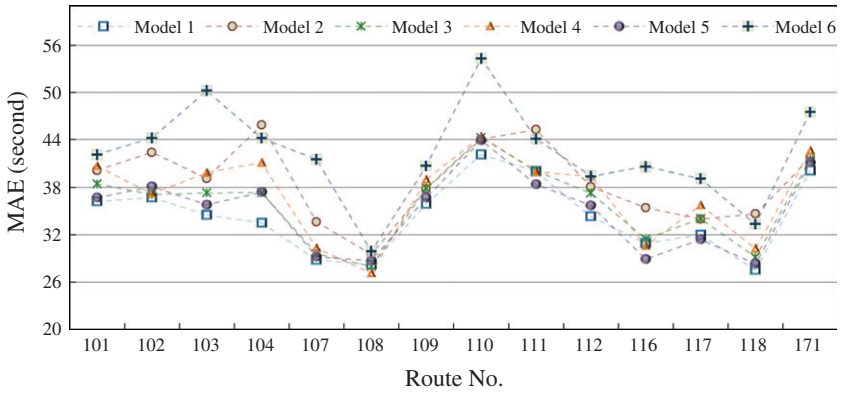


Fig. 9. Prediction errors of the ANN models with different input parameters.

models with different input parameters shown in Table 2 have also been trained by the BP algorithm. Fig. 9 shows the prediction errors using the six ANN models with different input parameters.

Similar to the comparison of the SVM models, the prediction error of the ANN model 6 (the single route prediction model) is the largest among the six ANN models. When comparing the average MAEs of the five ANN models (models 1 to 5) with bus information of multiple routes in Table 2, the ANN model 1 is the best model. However, it can also be observed from Table 2 and Fig. 9 that the prediction errors of the ANN models 1 and 5 are almost the same. To be consistence with the SVM model, the ANN model 5 with $\{t^l_{L,n}, t^i_{l,n}, \bar{t}^r_{L,n}, t^r_{l,n}\}$ is used for bus running time prediction in this study.

After determining the inputs of the ANN model, a scaled conjugate gradient algorithm (Moller, 1993) is used to train the ANN model. The number of hidden neurons is attained as five in this study. Thus, the final ANN model in this study is the ANN model with three-layer and five hidden neurons for bus running time prediction.

3.3.3. k nearest neighbours algorithm

In the *k*-NN model, the *k* nearest matches are determined among all the observations by the weighted distances. When computing the weighted distances, the parameters λ_1, λ_2, λ_3 and λ_4 are decided by the correlation coefficient of each input variable with the bus running time. In this study, $\lambda_1 = 0.268$, $\lambda_2 = 0.117$, $\lambda_3 = 0.217$ and $\lambda_4 = 0.348$, are calibrated respectively. To investigate the suitable value of *k* for the *k*-NN method in this study, sensitivity tests with different values of *k* in the range of 1–4 have been conducted. Fig. 10 shows that the prediction errors of the *k*-NN models with different value of *k*. Obviously, when *k* is set as 1, the prediction error is the largest among the four models. It is because that traffic conditions change rapidly and dynamically and accidents on the preceding bus will greatly affect the prediction accuracy. When *k* reaches 3 or 4, the prediction performance of the model is relatively better than that of the model with $k = 1$ or $k = 2$. It was also found that the difference of the prediction errors between the model with $k = 3$ or $k = 4$ is about 3%. Considering the complex of computation and the prediction accuracy together, $k = 3$ is selected for the *k*-NN model in this study.

3.3.4. Linear regression

Based on the preliminary analysis, it was found that time headways between the target bus and the last preceding bus of any route and of the same route ($t^l_{L,n}$ and $t^i_{l,n}$) are insignificant at 5% level for almost all the routes. Thus, these two variables

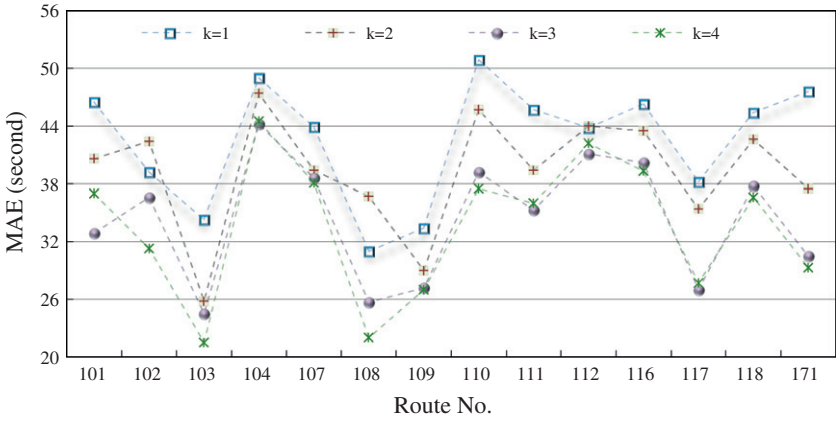


Fig. 10. Prediction errors of the *k*-NN models with different values of *k*.

Table 3
Coefficients of the independent variables in the LR model.

Route no.	$\ln(\bar{t}^r_{L,n})$	$\ln(t^r_{l,n})$	R^2	Route No.	$\ln(\bar{t}^r_{L,n})$	$\ln(t^r_{l,n})$	R^2
101	−0.210	1.222	0.82	103	−0.151	1.158	0.69
107	0.205	0.810*	0.74	104	0.353	0.646**	0.83
108	0.492	0.506	0.73	110	0.310	0.701**	0.79
109	1.122**	−0.123	0.82	112	0.269	0.735*	0.71
111	0.156	0.851*	0.67	117	0.446	0.557	0.84
116	0.143	0.856	0.77	118	0.176	0.831*	0.70
102	0.046	0.968*	0.81	171	0.073	0.931	0.83

* Significant at 0.05 level.
** Significant at 0.01 level.

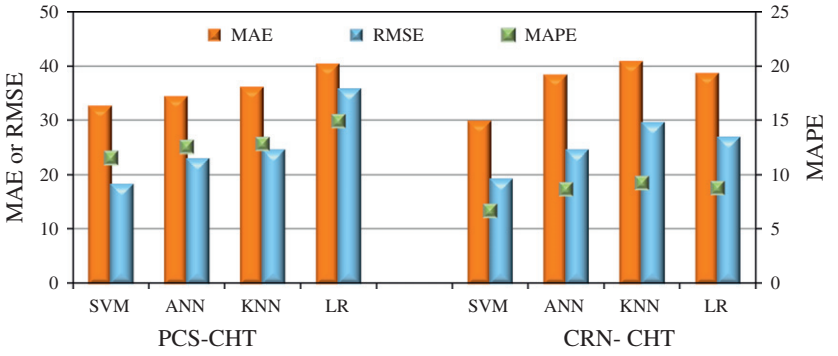


Fig. 11. Comparison of the performance among four methods.

were not adopted in the LR model. The coefficients of the independent variables in the resultant LR model for each of the bus route are shown in Table 3.

There is some difference in the coefficients of the two input variables in Table 3. $\ln(t^r_{l,n})$ shows higher influence on the predicted bus running time in most of the bus routes except the route no 109. This indicates that the running time of the preceding bus of the same route can provide more reliable information for the prediction. Furthermore, $\ln(t^r_{l,n})$ is significant at 5% level or 1% level for the route nos. 107, 111, 102, 104, 110, 112 and 118, while $\ln(\bar{t}^r_{L,n})$ is significant for the route No. 109.

3.4. Validation results

In this section, the bus arrival time at the CHT bus stop with multiple routes are forecasted by the SVM, ANN, *k*-NN and LR models. The average value of the MAE, the MAPE and the RMSE of the four models for all the bus routes are summarized in Fig. 11 and details are attached in Appendix A.

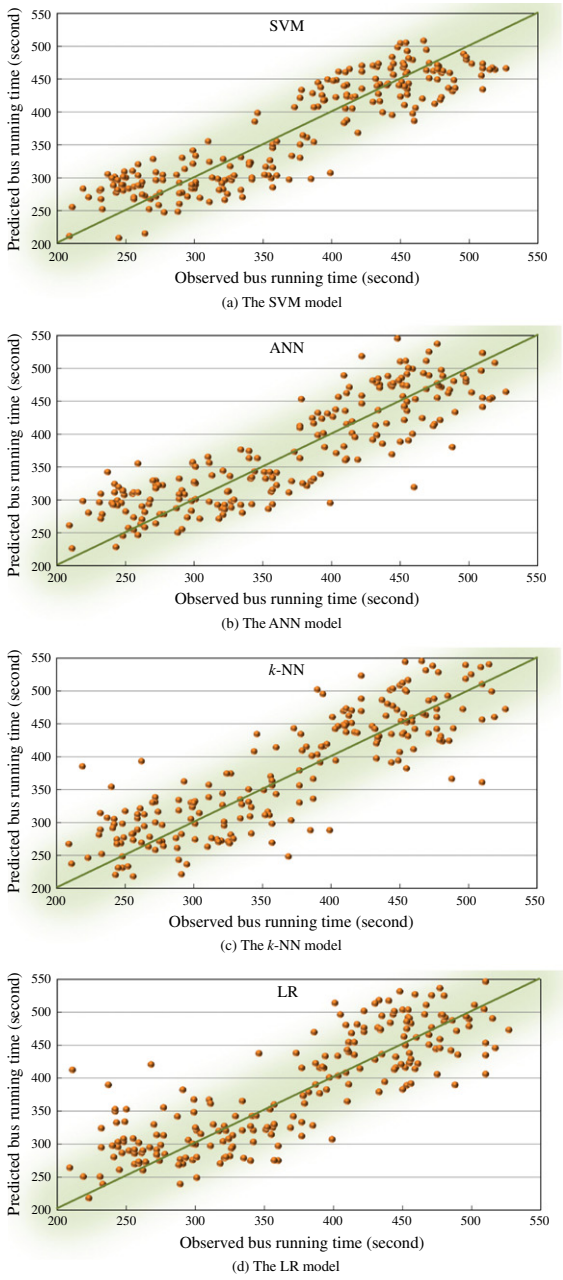


Fig. 12. Predictability of the four models on bus running time.

Table 4
The correlation coefficients (r) and the t-values for the four methods.

	SVM	ANN	k-NN	LR
r	0.9	0.87	0.85	0.84
t-value	−2.2	−1.98	1.57	1.54

Fig. 11 shows the comparison of the performance among four different methods from MAE, MAPE and RMSE, respectively. In Fig. 11, the horizontal axis is divided into two directions: PCS–CHT (the east direction) and CRN–CHT (the west direction). It can be seen that the SVM model has the best prediction performance among the four models in two directions. It is mainly due to that SVM implements the structural risk minimization principle and over-fitting is unlikely to occur with SVM.

Although the performance of the ANN model is worse than that of the SVM, the ANN model outperforms the *k*-NN and LR models. In summary, the performance of the LR model is worst among the four models. However, for the arrival time prediction for the west direction, the LR model is better than the *k*-NN model. From Appendix A, the average MAPEs of the SVM model are 11.5% and 6.69% for the east direction (from PCS to the CHT bus stop) and the west direction (from CRN to the CHT bus stop). For the fourteen bus routes, the MAPEs of the SVM model are from 4.49% to 13.23%, while the MAPEs of the ANN, *k*-NN and LR models are from 6.84% to 15.11%, from 6.94% to 16.89% and 6.78% to 24.99%, respectively. When comparing the maximum error of the prediction for each route using the different methods, it can be found that the maximum prediction error of the SVM model is the lowest except for the prediction for the route nos. 103 and 108. In summary, the SVM model has the best prediction performance among the four models for ten out of fourteen bus routes. Although the performance of the ANN model is slightly better than the one of the *k*-NN model, the *k*-NN model is still an alternative method for bus running time prediction due to its simple structure.

Fig. 12 and Table 4 show the bus running times predicted by the four models against the observed running times. It can be seen from the figure that the results of the SVM model are much closer to the observed data than the other three methods. The correlation coefficient (r) which reflects the accuracy of the bus running time prediction of the four methods is 0.90, 0.87, 0.85, 0.84, respectively. The coefficient implies that the proportion of the predicted bus running times from each model is well fitted with the observed bus running times. Also, it can be seen from the results of the *t*-tests of the four methods that only the SVM and ANN models passed the *t*-tests, whose *t*-values are larger than 1.96. This indicates that the SVM and ANN models are significant at 5% level. In summary, based on the validation results, the performance of the SVM model for bus arrival time prediction at the stop with multiple routes is shown to be satisfactory.

4. Conclusions

This paper investigated the bus arrival time prediction at bus stop with multiple routes. Bus running times of different routes were used to predict the bus arrival times by four proposed models, namely, SVM, ANN, *k*-NN and LR. In order to develop these four proposed models, bus running and arrival time data were collected from the observation surveys near the CHT in Kowloon urban area of Hong Kong. The results showed that the proposed models were more accurate than the models based on bus running times of single route. Moreover, the comparison results showed that the performance of the SVM model was the best among the four models for the bus arrival time prediction. It was also found that *k*-NN was an alternative method for the bus running time prediction compared with ANN. In summary, LR was the worst one among four models since its performance varied from the similarity between the current data and the preceding data, while compared with other three models, LR had the simplest structure.

In this paper, only the bus data was used to estimate the current traffic conditions. Further study will consider more factors such as running times of other vehicles (by type) and traffic flow variation so as to enhance the performance of the proposed prediction models.

Acknowledgements

The work described in this paper was jointly supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region to the Hong Kong Polytechnic University (Project No. PolyU 5195/07E), an internal research grant from the Research Committee of The Hong Kong Polytechnic University (Project No. J-BB7Q), the special grade of the financial support from China Postdoctoral Science Foundation 201003611 and Humanities and Social Sciences Foundation from the Ministry of Education of China 10YJC630357.

Appendix A. The details of results of four methods

Route no.		PCS–CHT							CRN–CHT								
		101	107	108	109	111	116	AVE	102	103	104	110	112	117	118	171	AVE
MAE (s)	SVM	39.4	34.9	28.7	25.2	34.2	32.6	32.5	32.9	24.4	29.4	34.8	32.6	21.3	26.2	36.5	29.8
	ANN	38.7	29.5	29.5	35.0	41.3	32.0	34.3	38.8	35.4	32.0	58.9	35.7	31.0	29.0	45.5	38.3
	k-NN	40.6	34.6	28.8	28.6	46.9	36.6	36.0	47.9	30.2	40.3	60.0	32.4	38.6	36.4	40.5	40.8
	LR	47.1	41.2	21.8	59.7	40.6	31.6	40.3	45.5	31.2	39.4	32.4	31.2	53.3	34.0	41.4	38.6
MAPE (%)	SVM	13.2	11.7	9.7	11.3	12.3	11.0	11.6	7.2	5.5	7.0	7.7	7.1	4.5	6.2	8.3	6.7
	ANN	12.8	11.1	10.4	14.5	15.1	11.4	12.6	8.6	8.4	7.4	13.0	7.8	6.9	6.8	10.5	8.7
	k-NN	13.6	10.8	10.0	12.7	16.9	13.1	12.8	10.4	7.3	9.5	13.6	6.9	8.6	8.7	9.1	9.3
	LR	16.8	14.7	7.8	25.0	14.1	11.2	14.9	10.1	7.6	9.1	7.4	6.8	11.7	8.3	9.3	8.8

(*continued on next page*)

Appendix A (*continued*)

Route no.		PCS–CHT							CRN–CHT								
		101	107	108	109	111	116	AVE	102	103	104	110	112	117	118	171	AVE
Max (s)	SVM	91.7	68.4	55.7	49.0	87.2	72.3	70.7	75.7	62.3	57.5	73.8	62.7	43.7	52.3	61.0	61.1
	ANN	103.8	104.7	55.0	74.0	96.3	77.1	85.2	97.6	79.8	107.7	141.0	65.8	44.0	62.0	96.3	86.8
	k-NN	120.6	70.4	48.4	58.2	165.6	81.5	90.8	149.2	59.7	122.1	130.6	64.4	60.0	108.4	101.0	99.4
	LR	107.3	152.5	51.4	201.4	110.5	91.9	119.2	83.4	71.4	98.4	83.0	69.3	101.2	112.6	104.1	90.4
RMSE	SVM	20.2	14.2	16.2	17.2	21.9	19.7	18.2	20.6	21.2	16.9	25.1	19.1	15.8	16.5	18.3	19.2
	ANN	27.6	27.8	13.4	24.0	23.2	21.6	22.9	28.3	23.8	24.2	42.1	20.5	12.1	21.9	23.2	24.5
	k-NN	32.2	23.8	14.8	16.5	37.1	22.4	24.5	37.7	22.3	32.7	38.0	21.3	25.8	30.5	28.0	29.5
	LR	31.8	39.8	17.6	73.7	25.9	25.5	35.7	20.2	23.0	30.0	27.2	21.3	30.8	32.8	29.5	26.8

References

Cathey, F.W., Dailey, D.J., 2003. A prescription for transit arrival/departure prediction using automatic vehicle location data. Transportation Research Part C 11 (3), 241–264.

Chan, K.S., Lam, W.H.K., Tam, M.L., 2009. Real-time estimation of arterial travel times with spatial travel time covariance relationships. Transportation Research Record 2121, 102–109.

Chang, H., Park, D., Lee, S., Lee, H., Baek, S., 2010. Dynamic multi-interval bus travel time prediction using bus transit data. Transportmetrica 6 (1), 19–38.

Chen, M., Liu, X., Xia, J., Chien, S.I., 2004. A dynamic bus arrival time prediction model based on APC data. Computer-Aided Civil and Infrastructure Engineering 19 (5), 364–376.

Chien, S.I.J., Kuchipudi, C.M., 2003. Dynamic travel time prediction with real-time and historic data. Journal of Transportation Engineering 129 (6), 608–616.

Chien, S.I.J., Ding, Y., Wei, C., 2002. Dynamic bus arrival time prediction with artificial neural networks. Journal of Transportation Engineering 128 (5), 429–438.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge, UK.

Ding, Y., Chien, S., 2000. The prediction of bus arrival times with link-based artificial neural networks. In: Proceedings of the Fifth Joint Conference on Information Sciences, Atlantic City, NJ.

Huang, S.H., Ran, B., 2003. An Application of Neural Network on Traffic Speed Prediction under Adverse Weather Condition. The 82nd Annual Meeting of the Transportation Research Board, Washington, DC.

Jeong, R., Rilett, L.R., 2004. Bus Arrival Time Prediction Using Artificial Neural Network Model. In: 7th International IEEE Conference on Intelligent Transportation Systems: (ITSC 2004), Washington DC.

Moller, M.F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6 (4), 523–533.

Park, S.H., Jeong, Y.J., Kim, T.J., 2007. Transit travel time forecasts for location-based queries: implementation and evaluation. Journal of the Eastern Asia Society for Transportation Studies 7, 1859–1869.

Shalaby, A., Farhan, A., 2004. Prediction models of bus arrival and departure times using AVL and APC data. Journal of Public Transportation 7 (1), 41–61.

Smith, B.L., Williams, B.M., Oswald, R.K., 2002. Comparison of parametric and nonparametric models for traffic flow forecasting. Transportation Research Part C 10 (4), 303–321.

Tam, M.L., Lam, W.H.K., 2008. Using automatic vehicle identification data for travel time estimation in Hong Kong. Transportmetrica 4 (3), 179–194.

Tam, M.L., Lam, W.H.K., 2009. Short-term Travel Time Prediction for Congested Urban Road Networks. The 88th Annual Meeting of the Transportation Research Board, Washington, DC.

van Hinsbergen, C.P.I.J., van Lint, J.W.C., van Zuylen, H.J., 2009. Bayesian committee of neural networks to predict travel times with confidence intervals. Transportation Research Part C 17(5), 498–509.

van Lint, J.W.C., Hoogendoorn, S.P., van Zuylen, H.J., 2005. Accurate freeway travel time prediction with state-space neural networks under missing data. Transportation Research Part C 13(5–6), 347–369.

Vapnik, V.N., 1999. An overview of statistical learning theory. IEEE Transactions on Neural Networks 10 (5), 988–999.

Vapnik, V.N., 2000. The Nature of Statistical Learning Theory. Springer, New York, NY, USA.

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transportation Research Part C 13 (3), 211–234.

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2006. Statistical methods for detecting nonlinearity and non-stationarity in univariate short-term time-series of traffic volume. Transportation Research Part C 14 (5), 351–367.

Wall, Z., Dailey, D.J., 1999. An algorithm for predicting the arrival time of mass transit vehicle using Automatic Vehicle Location data. The 78th Annual Meeting of the Transportation Research Board, Washington, DC.

You, J.S., Kim, T.J., 2000. Development and evaluation of a hybrid travel time forecasting model. Transportation Research Part C 8 (1–6), 231–256.

Yu, B., Yang, Z.Z., Yao, B.Z., 2006. Bus arrival time prediction using support vector machines. Journal of Intelligent Transportation Systems 10 (4), 151–158.

Yu, B., Yang, Z.Z., Chen, K., Yu, B., 2010a. A hybrid model for bus arrival time prediction. Journal of Advanced Transportation 44 (3), 193–204.

Yu, B., Yang, Z.Z., Wang, J., 2010b. Bus travel-time prediction based on bus speed. Proceedings of the Institution of Civil Engineers – Transport 163 (1), 3–7.

Zhang, X.Y., Rice, J.A., 2003. Short-term travel time prediction. Transportation Research Part C 11 (3–4), 187–210.

European Journal of Operational Research 196 (2009) 171–176

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing and Logistics

An improved ant colony optimization for vehicle routing problem

Yu Bin [a,*], Yang Zhong-Zhen [a], Yao Baozhen [b]

[a] College of Transportation and Logistics, Dalian Maritime University, Dalian 116026, PR China
[b] School of Civil Engineering and Architecture, Beijing Jiaotong University, Beijing 100044, PR China

ARTICLE INFO

Article history:
Received 8 August 2007
Accepted 22 February 2008
Available online 8 March 2008

Keywords:
Vehicle routing problem
Improved ant colony optimization
Ant-weight strategy
Mutation operation

ABSTRACT

The vehicle routing problem (VRP), a well-known combinatorial optimization problem, holds a central place in logistics management. This paper proposes an improved ant colony optimization (IACO), which possesses a new strategy to update the increased pheromone, called ant-weight strategy, and a mutation operation, to solve VRP. The computational results for fourteen benchmark problems are reported and compared to those of other metaheuristic approaches.

1. Introduction

Finding efficient vehicle routes is a representative logistics problem which has been studied for the last 40 years. A typical vehicle routing problem (VRP) aims to find a set of tours for several vehicles from a depot to a lot of customers and return to the depot without exceeding the capacity constraints of each vehicle at minimum cost. Since the customer combination is not restricted to the selection of vehicle routes, VRP is considered as a combinatorial optimization problem where the number of feasible solutions for the problem increases exponentially with the number of customers increasing (Bell and McMullen, 2004).

Heuristic algorithms such as simulated annealing (SA) (Chiang and Russell, 1996; Koulamas et al., 1994; Osman, 1993; Tavakkoli-Moghaddam et al., 2006), genetic algorithms (GAs) (Baker and Ayechew, 2003; Osman et al., 2005; Thangiah et al., 1994; Prins, 2004), tabu search (TS) (Gendreau et al., 1999; Semet and Taillard, 1993; Renaud et al., 1996; Brandao and Mercer, 1997; Osman, 1993) and ant colony optimization (Doerner et al., 2002; Reimann et al., 2002; Peng et al., 2005; Mazzeo and Loiseau, 2004; Bullnheimer et al., 1999; Doerner et al., 2004) are widely used for solving the VRP. Among these heuristic algorithms, ant colony optimizations (ACO) are new optimization methods proposed by Italian researchers Dorigo et al. (1996), which simulate the food-seeking behaviors of ant colonies in nature. It has been successfully applied as a solution to some classic compounding optimization problems, e.g. traveling salesman (Dorigo et al., 1996) quadratic assignment (Gambardella et al., 1997), job-shop scheduling (Colorni et al., 1994), telecommunication routing (Schoonderwoerd et al., 1997), etc.

If taking the central depot as the *nest* and customers as the *food*, the VRP is very similar to food-seeking behaviors of ant colonies in nature. This makes the coding of an ant colony optimization for the VRP is simple. Among the earliest studies was that of Bullnheimer et al. (1997) who presented a hybrid ant system algorithm with the 2-opt and the saving algorithm for the VRP. Other researches of the ACOs to the VRP included the work by Bullnheimer et al. (1999), Bell and McMullen (2004), Chen and Ting (2006). In the ACOs, the 2-opt exchange was used as an improvement heuristic within the routes found by individual vehicles and the pheromone updating rules mainly considered the global feature of the solution. This paper proposes an improved ant colony optimization with a new pheromone updating rule that can integrate the global feature and the local feature, a mutation operation and the 2-opt exchange for the VRP. The remainder of the paper is organized as follows. Section 2 presents the mathematical model for VRP. In Section 3, we present the IACO with ant-weight strategy and the mutation operation. Some computational results are discussed in Section 4 and lastly, the conclusions are provided in Section 5.

2. Vehicle routing problem

The VRP is described as a weighted graph $G = (C, L)$ where the nodes are represented by $C = \{c_0, c_1, \ldots, c_N\}$ and the arcs are represented by $L = \{(c_i, c_j): i \neq j\}$. In this graph model, c_0 is the central depot and the other nodes are the N customers to be served. Each node is associated with a fixed quantity q_i of goods to be delivered (a quantity $q_0 = 0$ is associated to the depot c_0). To each arc (c_i, c_j) is associated a value $d_{i,j}$ representing the distance between c_i and c_j.

* Corresponding author. Tel.: +86 411 84726756.
E-mail address: minlfish@yahoo.com.cn (B. Yu).

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2008.02.028

Each tour starts from and terminates at the depot c_0, each node c_i must be visited exactly once, and the quantity of goods to be delivered on a route should never exceed the vehicle capacity Q.

3. Improved ACO for VRP

3.1. Generation of solutions

Using ACO whose colony scale is P, an individual ant simulates a vehicle, and its route is constructed by incrementally selecting customers until all customers have been visited. The customers, who were already visited by an ant or violated its capacity constraints, are stored in the infeasible customer list (*tabu*).

The decision making about combining customers is based on a probabilistic rule taking into account both the visibility and the pheromone information. Thus, to select the next customer j for the kth ant at the ith node, the ant uses the following probabilistic formula.

$$p_{ij}(k) = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}}{\sum_{h \notin tabu_k} \tau_{ih}^{\alpha} \times \eta_{ih}^{\beta}} & j \notin tabu_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $p_{ij}(k)$ is the probability of choosing to combine customers i and j on the route, τ_{ij} the pheromone density of edge (i,j), η_{ij} the visibility of edge (i,j), α and β the relative influence of the pheromone trails and the visibility values, respectively and $tabu_k$ is the set of the infeasible nodes for the kth ant.

3.2. Mutation operation

Mutation operation referring to genetic algorithm (Yu and Yang, 2007; Yu et al., 2007) alters each child at a predefined probability. The operators can help the IACO to reach further solutions in the search space. The idea of the mutation operation is to randomly mutate the tour and hence produce a new solution that is not very far from the original one. In this paper, the mutation operator is designed to conduct customer exchanges in a random fashion. Fig. 2 shows the representation of the parent solution in Fig. 1. The steps for the mutation operation are as follows:

Step 1. Select the two tours from the selected parent solution and select the mutating point(s) from the each mutating tour. Fig. 3a shows the 9th customer in the 3rd tour and the 12th customer in the 4th tour are selected.

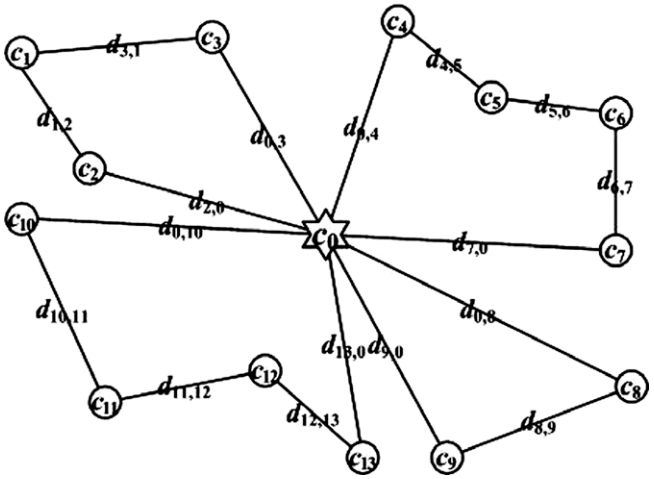


Fig. 1. An example of the VRP.

Parent solution

Route	Customer			
	3	1	2	
	4	5	6	7
	8	9		
	10	11	12	13

Fig. 2. An example of a parent solution.

Step 2. Exchange the customers in the different tours and generate the child solution (see Fig. 3b).

Step 3. Ensure the child solution local optimality. The 2-opt is applied to improve the mutated tours in child solution. Finally, the representation and the tours of the mutated child solution is as Figs. 3c and 4.

However, the mutation operation may violate vehicle capacity constraints. There are two approaches to deal with this situation. The first one is to assign a very high cost for such candidate solutions and accordingly reduce their probability of being selected in the forthcoming search. The second approach is to try to fix the resultant capacity violations by adjusting the delivery amounts. The advantage of the second approach over the first one is that it is more suitable in problems that are more likely to produce vehicle capacity violations and it enables IACO to investigate further points in the search space. Therefore, the second approach is adopted to deal with the vehicle capacity violation situation.

Each route of the solution is mutated with a certain probability p_m. Usually, the diversity of the solution is large at the beginning of a run and decreases with the time. We adapt the mutation rate during a run to promote a fast convergence to good solutions during the first generations and to introduce more diversity for escaping from local optima during later stages. The mutation probability at the generation t is

$$p_m(t) = p_m^{\min} + (p_m^{\max} - p_m^{\min})^{1-t/T} \quad (2)$$

where $p_m^{\min}$ and $p_m^{\max}$ are the lower and the upper mutation rates for the beginning and ending, respectively and T and t are the given maximum number of generations and the current generation of the iteration, respectively.

According to preliminary tests, we suggest to set the lower mutation rate to $p_m^{\min} = 1/n_c$, where n_c is the number of the customers, and the upper mutation rate to $p_m^{\max} = 1/n_v$, where n_v is the amounts of the routes in the solution.

Parent solution

Route	Customer			
	3	1	2	
	4	5	6	7
	8	9		
	10	11	12	13

Fig. 3a. Procedure of mutation (Step 1).

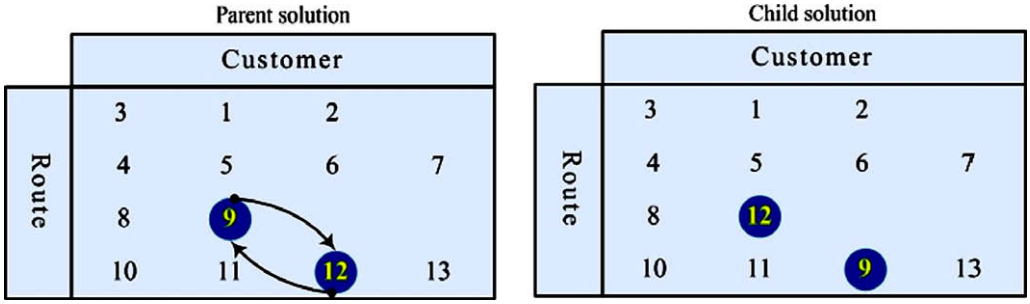


Fig. 3b. Procedure of mutation (Step 2).

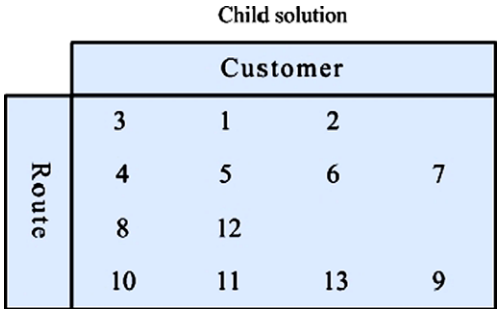


Fig. 3c. Procedure of mutation (Step 3).

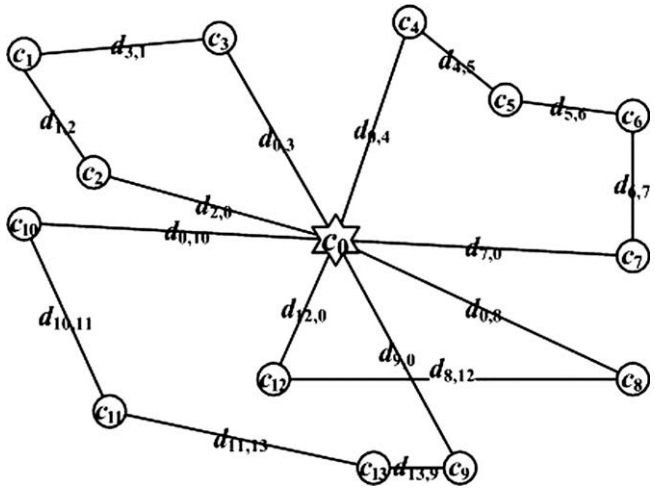


Fig. 4. Tours of the mutated child solution.

3.3. Local search

In the 2-opt exchange, all possible pairwise exchanges of customer locations visited by individual vehicles are tested to see if an overall improvement in the objective function can be attained. The method has been used in several ACOs (Bullnheimer et al., 1997, 1999; Bell and McMullen, 2004; Chen and Ting, 2006) for the VRP.

3.4. Update of pheromone information

The updating of the pheromone trails is a key element to the adaptive learning technique of ACO and the improvement of future solutions. First, Pheromone updating is conducted by reducing the amount of pheromone on all links in order to simulate the natural evaporation of the pheromone and to ensure that no one path becomes too dominant. This is done with the following pheromone updating equation,

$$\tau_{ij}^{\text{new}} = \rho \times \tau_{ij}^{\text{old}} + \sum_{k}^{K} \Delta\tau_{ij}^{k} \quad \rho \in (0,1) \tag{3}$$

where τ_{ij}^{new} is the pheromone on the link (i,j) after updating, τ_{ij}^{old} the pheromone on the link (i,j) before updating, ρ the constant that controls the speed of evaporation, k the number of the route, K the number of the routes in the solution and $K > 0$ and $\Delta\tau_{ij}^{k}$ are the increased pheromone on link (i,j) of route k found by the ant.

The pheromone increment updating rule uses the ant-weight strategy presented by Yang et al. (2007). Specifically, the strategy is written as:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{K\times L} \times \frac{D^k - d_{ij}}{m^k \times D^k} & \text{if link } (i,j) \text{ on the kth route} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where Q is a constant, L the total length of all routes in the solution, i.e. $L = \sum_k D^k$, D^k the length of the kth route in the solution, d_{ij} the length of edge (i,j) and m^k the number of customers in the kth routes and $m^k > 0$.

The ant-weight strategy updates the increased pheromone in terms of the solution quality and the contribution of each link to the solution, which consists of two components: the global pheromone increment and the local pheromone increment. In the ant-weight strategy, the quantity of the global pheromone increment, $Q/(K\times L)$, of each route is related to the total length of the solution, while the one of the local pheromone increment $(D^k - d_{ij})/(m^k \times D^k)$ of each link is based on the contribution of link (i,j) to the solution. Since the strategy for updating the increased pheromone considered both the global feature and local one of a solution, it can possibly ensure that the assigned increased pheromone is directly proportional to the quality of routes. The more favorable the link/route is the more pheromone increment is allocated to it, and the more accurate directive information is provided for later search. Meanwhile, by adjusting the pheromone assigning method for the links of current optimal path automatically, the algorithm can facilitate more delicate searches in the next cycle in a more favorable area, which assist in expanding the learning capacity from past searches. The parameters for updating the increased pheromone on the edges in the solution in Fig. 1 are calculated as Fig. 5.

Moreover, in order to prevent from local optimization and increase the probability of obtaining a higher-quality solution, upper and lower limits $[\tau_{\min}, \tau_{\max}]$ are fixed to the updating equation.

$$\tau_{\min} = Q \Big/ \sum_i 2d_{0i}, \tag{5}$$

$$\tau_{\max} = Q \Big/ \sum_i d_{0i}, \tag{6}$$

where d_{0i} is the distance from the central depot to the ith customer.

	Customer			
Route 1	3	1	2	
Route 2	4	5	6	7
Route 3	8	9		
Route 4	10	11	12	13

	D^k	m^k
Route 1	$D^1 = d_{0,3} + d_{3,1} + d_{1,2} + d_{2,0}$	3
Route 2	$D^2 = d_{0,4} + d_{4,5} + d_{5,6} + d_{6,7} + d_{7,0}$	4
Route 3	$D^3 = d_{0,8} + d_{8,9} + d_{9,0}$	2
Route 4	$D^4 = d_{0,10} + d_{10,11} + d_{11,12} + d_{12,13} + d_{13,0}$	4
$\sum$	$L = D^1 + D^2 + D^3 + D^4$	

	$\Delta\tau^k_{i,j}$		
Route 1	•••	$\Delta\tau^1_{3,1} = \frac{Q}{4\times L} \times \frac{D^1 - d_{3,1}}{3\times D^1}$	•••
Route 2	•••	•••	•••
Route 3	•••	•••	•••
Route 4	•••	$\Delta\tau^4_{10,11} = \frac{Q}{4\times L} \times \frac{D^4 - d_{10,11}}{4\times D^4}$	•••
$\sum$	$\sum_{ij}\Delta\tau^k_{ij} = \frac{Q}{4\times L}$		

Fig. 5. Parameters for updating the increased pheromone.

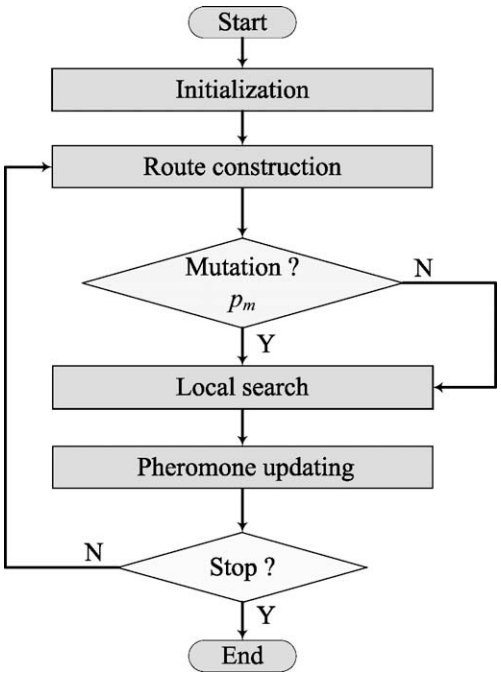


Fig. 6. The flowchart of IACO.

Table 1
Computational results using IACO and the well-known published results

No.	n	Q	Best know	Best	Worst	Average	Time
C1	50	160	524.61[a]	**524.61**	524.61	524.61	2
C2	75	140	835.26[a]	**835.26**	859.3	848.85	11
C3	100	200	826.14[a]	830.00	861.12	844.32	30
C4	150	200	1028.42[a]	**1028.42**	1067.1	1042.52	211
C5	199	200	1291.45[b]	1305.5	1344.41	1321.91	677
C6	50	160	555.43[a]	**555.43**	568.89	560.14	24
C7	75	140	909.68[a]	**909.68**	942.29	919.1	20
C8	100	200	865.94[a]	**865.94**	888.89	871.52	57
C9	150	200	1162.55[a]	**1162.55**	1228.9	1194.87	307
C10	199	200	1395.85[b]	**1395.85**	1433.68	1412.92	840
C11	120	200	1042.11[a]	**1042.11**	1056.26	1048.12	61
C12	100	200	819.56[a]	**819.56**	842.51	823.66	31
C13	120	200	1541.14[a]	1545.93	1572.29	1552.25	127
C14	100	200	866.37[a]	**866.37**	869.12	867.05	43

[a] Taillard (1993).
[b] Rochat and Taillard (1995).

3.5. Overall procedure

The flowchart of our IACO for the VRP is shown in Fig. 6.

4. Numerical analysis

The heuristics described in the previous sections is applied to the 14 vehicle routing problems which can be downloaded from the OR-library (see Beasley, 1990), and which have been widely used as benchmarks, in order to compare its ability to find the solution to VRP. The information of the 14 problems is shown in columns 2–4 in Table 1, which consists of the problem size n, the vehicle capacity Q, and the well-known published results (Taillard, 1993; Rochat and Taillard, 1995). The IACO parameters used for VRP instances are $Q = 1000$, $\alpha = 2$, $\beta = 1$ and $\rho = 0.8$. Then, the IACO were coded in Visual C++.Net 2003 and executed on a PC equipped with 512 MB of RAM and a Pentium processor running at 1000 MHz. Columns 5–8 present the results from IACO including the best solution, the worst solution, and the average solution and average run time (second). The numbers in bold are the results as the best-known solutions.

To evaluate the ant-weight strategy and the mutation operation, the two ant colony optimizations with different strategy are constructed. The first one is a standard ant colony optimization with the ant-weight strategy (denoted by ACO-W) and the other is a standard ant colony optimization with the mutation operation (denoted by ACO-M). The computational results are as shown in Table 2. The numbers in bold are the best solutions among three algorithms. It can be observed that ACO-M can obtain the same solutions as IACO in test problem 1, 2, 3, 6, 7, 11, 12 and 14, while ACO-W can only obtain the optimum solutions in test problem 1, 2, 6, 7 and 14. Compared with ACO-W, the ACO-M generally provides better solutions for the 14 problems. This may be attributed that the introduction of the mutation operation can diversify the ant colony, explore new possible solution space and prevent the algorithm from trapping in local optimization. However, while the mutation operation improves the solutions, it also increases the computation times. We can see that the times consumed by

Table 2
Computational results using IACO, ACO-W and ACO-M

No.	IACO				ACO-W				ACO-M			
	Best	Worst	Average	Time	Best	Worst	Average	Time	Best	Worst	Average	Time
C1	**524.61**	524.61	524.61	2	**524.61**	524.61	524.61	2	**524.61**	524.61	524.61	2
C2	**835.26**	859.3	848.85	11	**838.25**	859.3	850.05	10	**835.26**	859.3	849.33	13
C3	**830.00**	861.12	844.32	30	834.36	861.12	851.02	27	**830.00**	861.12	847.26	51
C4	**1028.42**	1067.1	1042.52	211	1044.89	1087.3	1058.62	198	1033.26	1084.31	1052.52	584
C5	**1305.5**	1344.41	1321.91	677	1335.36	1387.85	1362.37	602	1310.21	1377.29	1340.07	1,134
C6	**555.43**	568.89	560.14	24	**555.43**	571.17	563.32	24	**555.43**	577.49	564.18	24
C7	**909.68**	942.29	919.1	20	**909.68**	947.87	927.06	19	**909.68**	950.1	931.07	22
C8	**865.94**	888.89	871.52	57	876.52	901.06	888.87	52	869.91	901.06	880.03	79
C9	**1162.55**	1228.9	1194.87	307	1204.47	1279.39	1252.05	292	1188	1253.31	1222.24	772
C10	**1395.85**	1433.68	1412.92	840	1439.07	1520.04	1487.78	780	1412.12	1492.36	1466.62	1,320
C11	**1042.11**	1056.26	1048.12	61	1051.71	1077.33	1059.13	55	**1042.11**	1060.61	1055.52	204
C12	**819.56**	842.51	823.66	31	833.31	850.04	840.61	30	**819.56**	851.11	844.06	77
C13	**1545.93**	1572.29	1552.25	127	1571.05	1622.88	1592.83	118	1556.86	1618.58	1588.21	402
C14	**866.37**	869.12	867.05	43	**866.37**	870.18	868.09	38	**866.37**	870.18	868.61	79

ACO-M are more than ACO-W. This may be because that in ACO-W, the ant-weight strategy is used for updating the increased pheromone. It can assign increased pheromone according to the quality the solution. This improves the learning capacity of the algorithm from past searches, and enhances the efficiency. Furthermore, IACO integrated the ant-weight and the mutation operation can provide the best solutions and consume the less computation times compared with ACO-W and ACO-M.

Table 3
Deviations from the best known solution of several metaheuristic approaches

Prob.	RR-PTS	G-TS	OSM-TS	OSM-SA	B-AS	IACO
C1	**0.00**	**0.00**	**0.00**	0.65	**0.00**	**0.00**
C2	0.01	0.06	1.05	0.40	1.08	**0.00**
C3	**0.17**	0.40	1.44	0.37	0.75	0.47
C4	1.55	0.75	1.55	2.88	3.22	**0.00**
C5	3.34	2.42	3.31	6.55	4.03	**1.08**
C6	**0.00**	**0.00**	**0.00**	**0.00**	0.87	**0.00**
C7	**0.00**	0.39	0.15	**0.00**	0.72	**0.00**
C8	0.09	0.00	1.39	0.09	0.09	**0.00**
C9	0.14	1.31	1.85	0.14	2.88	**0.00**
C10	1.79	1.62	3.23	1.58	4.00	**0.00**
C11	**0.00**	3.01	0.09	12.85	2.22	**0.00**
C12	**0.00**	**0.00**	0.01	0.79	**0.00**	**0.00**
C13	0.59	2.12	0.31	**0.31**	1.22	**0.31**
C14	**0.00**	**0.00**	**0.00**	2.73	0.08	**0.00**
Average	0.55	0.86	1.03	2.10	1.51	0.14

Our IACO are compared with five other meta-heuristic approaches in the paper proposed by Bullnheimer et al. (1997), which consisted of parallel tabu search algorithm (RR-PTS) by Rego and Roucairol, a tabu search algorithm (G-TS) by Gendreau et al., tabu search (OSM-TS), a simulated annealing algorithm (OSM-SA) by Osman and ant system algorithm (B-AS) by Bullnheimer et al. (1997). The comparison of the deviations from the best known solution is shown in Table 3. The performance of our IACO is best among all meta-heuristic approaches, who produces in eleven problems of fourteen test problems and yields the lowest average deviation. Also, compared with OSM-SA, the tabu search approaches are able to provide better solutions. Also, compared OSM-SA, the tabu search approaches can provide better solutions.

For a correct evaluation and comparison of the quality of six algorithms the computing times must be taken into account. However, a correct evaluation and comparison of the computing times is generally tough due to the enormous variety of computers available and used by different researchers. A very rough measure of computers' performance can be obtained using Dongarra's (Dongarra, 2001) tables where the number (in millions) of floating-point operations per second (Mflop/seconds) executed by each computer was used, when solving standard linear equations, with LINPACK program. Regarding computational times, Rego and Roucairol used a sun sparc 4 (about 5.7 MFlop/s), Gendreau et al. used a 36 MHz Silicon Graphics (about 6.7 MFlop/s), Osman used a VAX 8600(about 2.48 MFlop/s), Bullnheimer et al. used a Pentium 100 MHz (about 8 MFlop/s). In this research, the Pentium 1 GHz running IACO has an estimated power of 75 MFlop/s. Table 4 shows

Table 4
Computation times of several metaheuristic approaches

Probability	RR-PTS		G-TS		OSM-TS		OSM-SA		B-AS		IACO
	Run times	Scaled times	Run times	Scaled times	Run times	Scaled times	Run times	Scaled times	Run times	Scaled times	Run times
C1	66	5.0	84	7.5	60	2.0	6	0.2	6	0.6	2
C2	2604	197.9	2352	210.2	48	1.5	3564	117.8	78	8.3	11
C3	1578	119.9	408	36.4	894	29.5	6174	204.1	228	24.3	30
C4	2910	221.2	3270	292.1	1764	58.3	4296	142.1	1104	117.8	211
C5	4626	351.6	5028	449.1	1704	56.3	1374	45.4	5256	560.6	677
C6	144	10.9	468	41.8	60	2.0	696	23.0	6	0.6	24
C7	1236	94.0	1908	170.4	744	24.6	312	10.4	102	10.9	20
C8	1134	86.2	354	31.6	1962	64.8	366	12.1	288	30.7	57
C9	1794	136.3	1278	114.1	2472	81.8	59,016	1951.5	1650	176.0	307
C10	2562	194.8	2646	236.4	4026	133.0	2418	79.9	4908	523.5	840
C11	672	51.1	714	63.8	780	25.8	264	8.7	552	58.9	61
C12	96	7.3	102	9.2	342	11.2	48	1.5	300	32.0	31
C13	120	9.1	2088	186.5	1578	52.1	4572	151.1	660	70.4	127
C14	1482	112.7	1782	159.2	582	19.3	300	9.9	348	37.1	43
Average	–	114.14	–	143.46	–	40.17	–	196.98	–	117.98	172.93

the origin computation times and the scaled computation times, which use Pentium 1 GHz as the baseline, of six approaches.

The performance of IACO is competitive when compared with other meta-heuristic approaches, such as SA, and TS. Although the run times are not favor in IACO, our IACO still seems to be superior in terms of solution quality with an average deviation of 0.14%. Considering the very rough measure, the scaled times are viewed as the assistant aspect of the performance. Regarding the computation efficiency, we find that the IACO can find very good solutions in an acceptable time.

5. Conclusions

The VRP has been an important problem in the field of distribution and logistics. Since the delivery routs consist of any combination of customers, this problem belongs to the class of NP-hard problems. This paper presents an IACO with ant-weight strategy and a mutation operation. The computational results of 14 benchmark problems reveal that the proposed IACO is effective and efficient. Further research on additional modifications of the IACO to extensions of the vehicle routing problem with time windows or with more depots, are of interest.

References

Baker, B.M., Ayechew, M.A., 2003. A genetic algorithm for the vehicle routing problem. Computers & Operations Research 30, 787–800.

Beasley, J.E., 1990. OR-Library: distributing test problems by electronic mail. Journal of the Operational Research Society 41, 1069–1072.

Bell, J.E., McMullen, P.R., 2004. Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics 1 (8), 41–48.

Brandao, J., Mercer, A., 1997. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. European Journal of Operational Research 100, 180–191.

Bullnheimer, B., Hartl, R.F., Strauss, C., 1997. Applying the ant system to the vehicle routing problem. In: Second Metaheuristics International Conference, MIC'97, Sophia-Antipolis, France.

Bullnheimer, B., Hartl, R.F., Strauss, C., 1999. An improved ant system algorithm for the vehicle routing problem. Annals of Operations Research 89, 319–328.

Chen, C.H., Ting, C.J., 2006. An improved ant colony system algorithm for the vehicle routing problem. Journal of the Chinese Institute of Industrial Engineers 23 (2), 115–126.

Chiang, W.C., Russell, R., 1996. Simulated annealing meta-heuristics for the vehicle routing problem with time windows. Annals of Operations Research 93, 3–27.

Colorni, A., Dorigo, M., Maniezzo, V., Trubian, M., 1994. Ant system for job-shop scheduling. Jorbel–Belgian Journal of Operations Research Statistics and Computer Science 34 (1), 39–53.

Doerner, K.F., Gronalt, M., Hartl, R.F., Reimann, M., Strauss, C., Stummer, M., 2002. Savings ants for the vehicle routing problem. Applications of Evolutionary Computing. Springer, Berlin.

Doerner, K.F., Hartl, R.F., Kiechle, G., Lucka, M., Reimann, M., 2004. Parallel ant systems for the capacitated vehicle routing problem. In: Evolutionary Computation in Combinatorial Optimization: 4th European Conference, EvoCOP 2004, LNCS 3004, pp. 72–83.

Dongarra, J., 2001. Performance of various computer using standard linear equations software. Report CS-89-85, University of Tennessee.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Mans, and Cybernetics 1 (26), 29–41.

Gambardella, L., Taillard, E., Dorigo, M., 1997. Ant Colonies for the QAP, Technical Report 97-4, IDSIA, Lugano, Switzerland.

Gendreau, M., Laporte, G., Musaraganyi, C., Taillard, E.D., 1999. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers & Operations Research 26, 1153–1173.

Koulamas, C., Antony, S., Jaen, R., 1994. A survey of simulated annealing applications to operations research problems. Omega 22 (1), 41–56.

Mazzeo, S., Loiseau, I., 2004. An ant colony algorithm for the capacitated vehicle routing. Electronic Notes in Discrete Mathematics 18, 181–186.

Osman, I.H., 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research 41, 421–451.

Osman, M.S., Abo-Sinna, M.A., Mousa, A.A., 2005. An effective genetic algorithm approach to multiobjective routing problems (morps). Applied Mathematics and Computation 163, 769–781.

Peng, W., Tong, R.F., Tang, M., Dong, J.X., 2005. Ant colony search algorithms for optimal packing problem. ICNC 2005, LNCS 3611, pp. 1229–1238.

Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research 31, 1985–2002.

Reimann, M., Stummer, M., Doerner, K., 2002. A savings based ant system for the vehicle routing problem. In: Langdon, W.B. et al. (Eds.), GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann, San Francisco.

Renaud, J., Laporte, G., Boctor, F.F., 1996. A tabu search heuristic for the multi-depot vehicle routing problem. Computers & Operations Research 23 (3), 229–235.

Rochat, Y., Taillard, R.E., 1995. Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics 1, 147–167.

Schoonderwoerd, R., Holland, O., Bruten, J., Rothkrantz, L., 1997. Ant-based load balancing in telecommunications networks. Adaptive Behavior 5 (2), 169–207.

Semet, F., Taillard, E.D., 1993. Solving real-life vehicle routing problems efficiently using taboo search. Annals of Operations Research 41, 469–488.

Taillard, R.E., 1993. Parallel iterative search methods for vehicle routing problems. Networks 23, 661–673.

Tavakkoli-Moghaddam, R., Safaei, N., Gholipour, Y., 2006. A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. Applied Mathematics and Computation 176, 445–454.

Thangiah, S.R., Osman, I.H., Sun, T., 1994. Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows. Technical Report 27, Computer Science Department, Slippery Rock University.

Yang, Z.Z., Yu, B., Cheng, C.T., 2007. A parallel ant colony algorithm for bus network optimization. Computer-Aided Civil and Infrastructure Engineering 22, 44–55.

Yu, B., Yang, Z.Z., 2007. A dynamic holding strategy in public transit systems with real-time information. Applied Intelligence, (accepted for publication), doi:10.1007/s10489-007-0112-9.

Yu, B., Yang, Z.Z., Cheng, C.T., 2007. Optimizing the distribution of shopping centers with parallel genetic algorithm. Engineering Applications of Artificial Intelligence 20 (2), 215–223.