



# A knowledge based real-time travel time prediction system for urban network

Wei-Hsun Lee<sup>a,b,\*</sup>, Shian-Shyong Tseng<sup>a,c</sup>, Sheng-Han Tsai<sup>a</sup>

<sup>a</sup> Department of Computer Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan, ROC

<sup>b</sup> Telecommunication Laboratories, Chunghwa Co. Ltd.

<sup>c</sup> Department of Information Science and Applications, Asia University, 500 Liufeng Road, Wufeng, Taichung 413, Taiwan, ROC

## ARTICLE INFO

### Keywords:

Knowledge based system  
Spatiotemporal data mining  
Travel time prediction  
Intelligent transportation system (ITS)

## ABSTRACT

Many approaches had been proposed for travel time prediction in these decades; most of them focus on the predicting the travel time on freeway or simple arterial network. Travel time prediction for urban network in real time is hard to achieve for several reasons: complexity and path routing problem in urban network, unavailability of real-time sensor data, spatiotemporal data coverage problem, and lacking real-time events consideration. In this paper, we propose a knowledge based real-time travel time prediction model which contains real-time and historical travel time predictors to discover traffic patterns from the raw data of location based services by data mining technique and transform them to travel time prediction rules. Besides, dynamic weight combination of the two predictors by meta-rules is proposed to provide a real-time traffic event response mechanism to enhance the precision of the travel time prediction.

© 2008 Elsevier Ltd. All rights reserved.

## 1. Introduction

Nowadays, travel time information plays an important role in several fields of intelligent transportation systems (ITS), such as advanced traffic management systems (ATMS), advanced traveler information system (ATIS), commercial vehicle operation (CVO) and emergency management system (EMS). Besides, travel time prediction (TTP) also contributes to traveler, traffic administrator, and logistics operators. For travelers and logistic operators, accurate travel time estimation could avoid congested sections to reduce transport costs and increase service quality. For traffic managers, travel time information is an important index of traffic system operation. Furthermore, using travel time information can scatter the condensed traffic volume and sharply reduce the habitual traffic congestion in effective, because people might choose various public transportations as their wishes. So, real-time TTP is a meaningful traffic index to be referred. However, TTP for urban network is highly stochastic and time-dependent due to random fluctuation in travel demands, interruptions caused by traffic control devices, incidents, road construction, and weather conditions. In other words, travel time is affected by a set of traffic factors including speed limit, traffic volume, routing path selected, occupancy of road and traffic facilities (e.g., signals) as well as non-traffic factors including traffic event, weather, road construction, etc. Most previous researches predicted travel time based on some traffic factors, such as speed, volume or occupancy, and did not take

the non-traffic factors into consideration. Thus, the previous research approaches did not work well in real traffic condition.

Study by Iroy and Kuwahara (2001) found that level of reduction in congestion depends on the complexity of the road network. Vehicular flows on freeways are often treated as uninterrupted flows; flows on urban network are conceivably much more complicated since vehicles traveling on urban network are subject to not only queuing delays but also signal delays as well as to turning delays. Thus, TTP for an urban network is more challenging than predicting the travel time for freeway or single arterial. Besides, the routing and path selection problems should be solved in TTP for urban network, e.g., the TTP model has to decide which path on a given OD (origin and destination) pair as request to be the suggested path. Many models had been proposed for travel time prediction in these decades, but most of them focused on predicting the travel time on freeway (Chien & Kuchipudi, 2003; Rice & van Zwet, 2004; Wu, Ho, & Lee, 2004) or simple arterial network (Jiang & Zhang, 2003; Lin, Kulkarni, & Mirchandani, 2004).

In the past, many ITS studies and transportation agencies use the traffic data from dual-loop detectors which are capable of archiving with traffic count (the number of vehicles that pass over the detector in that period of time), velocity, and occupancy (the fraction of time that vehicles are detected) and readily available in many locales of freeways and urban roadways (Lin & Zito, 2005). Nowadays, traffic data collecting techniques have made great progress and evolved to real-time collecting in order to improve traffic management efficiency. In Lin and Zito (2005), traffic information collection and travel time measurement can be divided into three categories: site-based, vehicle-based and sensor-based measurement. Site-based measurement collects vehicle

\* Corresponding author. Address: Department of Computer Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan, ROC.  
E-mail address: [leews@cht.com.tw](mailto:leews@cht.com.tw) (W.-H. Lee).

license plate characters and arrival times at various checkpoints through automatic vehicle identification (AVI) technologies, matches the license plates between consecutive checkpoints, and computes travel times from the difference between arrival times. Vehicle-based methods make TTP by analyzing raw data collected from fleet of probe vehicles. Sensor-based methods make TTP measurement by collecting raw data from the stationary sensors like loops detectors, transponders or radio beacons installed at arterial roads. However, each traffic information collection method used for TTP has some drawbacks and limitations. For example, site-based and sensor-based TTP methods have the spatial coverage problem because the sensors or AVI devices are fixed and limited. Vehicle-based TTP methods (Chung, Sarvi, Murakami, Horiguchi, & Kuwahara, 2003; Nakata & Takeuchi, 2004; Yang, 2005) have cost, spatial and temporal coverage problems because the total cost is very high if a dedicated fleet of urban network traffic probing vehicle is maintained.

There are numerous previous TTP approaches based on the historical traffic data analysis in the literatures, which can be categorized as follows (Lin & Zito, 2005): regression method (mathematical model) (Wu et al., 2004), time series estimation method, hybrid of data fusion or combinative model (Wen, Lee, & Cho, 2005) and artificial intelligence method like neural network (Mark, Sadek, & Rizzo, 2004). In Nakata and Takeuchi (2004), auto regression (AR) model and state space model for time series modeling were used to predict travel time. The Kalman filtering provides an efficient computational (recursive) in many TTP researches (Chung, 2003; Chung et al., 2003; Lin et al., 2004; Yang, 2005), because it is very powerful in several aspects: it supports estimations of past, present, and even future states even if the precise nature of the modeled system is unknown. In Wu et al. (2004), the support vector regression model was used to predict travel time for highway users. In Bajwa, Chung, and Kuwahara (2004), pattern matching technique was used for TTP. Traffic patterns similar to the current traffic are searched among the historical patterns, and the closest matched patterns are used to extrapolate the present traffic condition. Chung et al. (2003) developed an OD estimation method to make more accurate estimation of traffic flow and traffic volume in congestion traffic status. Moreover, the data fusion models of TTP integrated grey theory (Takahashi, Takahashi, & Izumi, 2003) and neural network-based. Yang (2005) developed some hybrid models toward data treatment and data fusion for traffic detector data on freeway.

However, most of the previous works only consider the static models of spatial network and predicted travel time based on the historical collected data, and thus lack the consideration of real-time events and traffic status. In other words, none of real-time events (e.g., detours and traffic congestions) affecting the spatial network can be reflected in the prediction result. Travel time prediction for urban network in real time is hard to achieve due to the following reasons: (1) network complexity, (2) path routing and selection problem in road network, (3) the collection of sensor data in real time is not available or cost-effective, (4) spatiotemporal data coverage problem of sensor- or vehicle-based travel time prediction, and (5) low precision due to lack of event response mechanism. To make TTP system more practical and more precise, a real-time knowledge based TTP model is proposed in this paper to take the advantage of independent knowledge base so that TTP knowledge can be dynamically adjusted to fit the changing requirement and response to external events. It results in that prior knowledge contributed by domain expert (meta-rules) and pattern knowledge mining from LBS-based applications can be evolved with the environment.

The basic idea of the proposed TTP model is that travel time along a selected path can be estimated by summing up the links travel time with intersections delay, as shown in Eq. (1), where link

travel time can be estimated by linear combination of current (real time) and historical predictors. Origin ( $O$ ), destination ( $D$ ) and journey start time ( $t$ ) are the input parameters of the prediction formula  $T(O, D, t)$ . Two sub-functions  $T_c$  and  $T_h$  are two link travel time predictors based on current and historical traffic information, respectively, and  $T_d$  presents the total intersection delays of the passing through intersections in the path. Two control variables  $\alpha$  and  $\beta$  are the weighted combination variables for current ( $T_c$ ) and historical ( $T_h$ ) predictors, respectively. The weights for these two predictors are decided by meta-rules given by domain expert, which are discussed in Section 3.

$$T(O, D, t) = \alpha \cdot T_c(O, D) + \beta \cdot T_h(O, D, t) + T_d(O, D, t) \quad (1)$$

where  $\alpha + \beta = 1$

The objective of this paper is to propose a real-time TTP model for an urban network that predicts travel time by linear combination of the results of real-time and historical travel time predictors based on the request of an origin ( $O$ ) and destination ( $D$ ) pair. The model utilizes the raw data of location-based services (LBS), transforms it to the traffic information by combining the geographical information system (GIS), and predicts travel time by integrating the historical traffic data regression, real-time traffic information and real-time external information sources, where external information sources include the real-time information that may affect TTP, such as incidents, road construction, and weather. Besides, meta-rules offered by traffic domain experts to raise the precision of real-time TTP can dynamically tune the combination weights of historical and real-time TTP according to the external events. For example, a current car accident on a link in the path of  $O, D$  pair may trigger one of the meta-rules to raise the weight of real-time TTP on that link, because the traffic delay on that link will be reflected immediately by the real-time LBS. This model combining the data mining and knowledge based system technologies mines the traffic patterns from location-based services (LBS) and transforms them to the TTP inference rules, so that it can handle the issues of non-traffic factors as well as traffic factors.

The model proposed in this paper utilizes the raw data of LBS-based applications, and regards the vehicles in the LBS-based applications as the traffic probing vehicle. Comparing to the traditional vehicle-based TTP, it is cost-effective because the traffic information is derived by only mining the raw data of LBS-based applications. Moreover, the size of LBS fleet has the temporal and spatial coverage advantages. Traffic information can be dynamically gathered in the LBS fleet operation area 24 h per day in real time.

The following sections are arranged as follows. Section 2 gives the introduction of LBS, and explains how traffic information can be derived from LBS. The proposed knowledge based TTP methodology is detailedly discussed in Section 3. In Section 4, we implemented the prototype TTP system for Taipei urban network by utilizing the taxi dispatching system as our LBS data source. Real-time, historical and linear combination predictors are evaluated and compared in this section. Finally, concluding remarks and future research are presented in Section 5.

## 2. Traffic information derived from LBS

LBS, providing appropriate location aware information for the users in different locations through the mobile communication network, has become the main stream of mobile commerce applications and telematics services. There are various kinds of LBS-based applications. For examples, vehicle positioning system (VPS) for electronic toll collection (Lee, Jeng, Tseng, & Wang, 2004), taxi dispatching system (TDS) (Liu, Wang, Shieh, & Jeng, 2004), commercial fleet management systems, and vehicle security

systems. The common architecture of the LBS-based system includes four parts: on-board units (OBU), communication system (cellular network), and backend systems (M-server and E-server) as shown in Fig. 1, where OBU is a small computer system which is installed on the vehicle with computing, positioning, communication, and human interface modules and the communication system which can be any wireless communication mechanism, such as GSM/GPRS/UMTS cellular network and 802.11/16 wireless network, is the link between OBU and the backend system. Till now, GPRS cellular network is the most popular communication system in commercialized system.

The backend system consists of two parts: M-server and E-server. M-server is responsible for transmitting bi-direction messages and serves as a buffer of uplink and downlink packets between OBUs and backend system over the mobile network. E-server consists of GIS engine, database and application server which keeps the positions and status (e.g., speed and state) of all the vehicles by collecting the uplink reports of OBU and is responsible for the information processes of all the business workflow according to the business rules in LBS applications.

The vehicles in LBS are regarded as the traffic status probing vehicles of the urban network, where a vehicle in the LBS is equipped with an OBU, which has global positioning system (GPS) positioning module and mobile data communication module such as GPRS/UMTS to be able to report vehicle position, traveling direction, and speed from the GPS module and uplink the vehicle status to the backend system through mobile communication module.

$$U_p(X, Y, t, V, D, S) \xrightarrow{\text{GIS}} \text{TIS}(L_i, T_s, V, D) \quad (2)$$

Each uplink packet ( $U_p$ ), as shown in Eq. (2), being sent from the probing vehicle to the backend system, represents the current position and traveling status of that vehicle. The information in  $U_p$  includes: position coordinate ( $X, Y$ ), traveling speed ( $V$ ), direction ( $D$ ), timestamp ( $t$ ) and status ( $S$ ). By combining road network database in GIS, the coordinate of a vehicle can be transformed to nearest address by interpolating the GPS position with road network database (Wang et al., 2003). Thus traffic information can be gathered by transforming the uplink packet into traffic information spot (TIS) of the link spatial dimension ( $L_i$ ) and temporal dimension ( $T_s$ ) where the vehicle located at  $L_i$ , as shown in Eq. (2), has the traveling speed ( $V$ ) and direction ( $D$ ). Then, the real-time traffic information of the urban network can be derived from LBS by aggregating all the collected TISs at the current-time interval, for example, half an hour.

Besides the real-time traffic information, historical traffic database also plays an important role in TTP. Historical traffic information is stored in the format of journey set, where a journey which represents the tracks of the vehicle starting from the origin to the destination is a collection of consecutive TISs of a vehicle. All the

TISs in the journey set are classified by temporal and spatial condition rules, and then stored into the historical traffic information database. Traffic patterns and rules for TTP can be extracted from the database by data mining technology (Tsai, Lee, & Tseng, 2005), and then be transformed into knowledge rules.

### 3. Knowledge based travel time prediction

The proposed real-time knowledge based TTP model predicts travel time based on the knowledge based system and data mining technologies. There are two categories of knowledge in this model: (a) general rules for real-time and historical TTP are obtained by mining the LBS-based applications and (b) meta-rules donated by the human domain experts. Meta-rules play a key position in the TTP model for following four purposes:

- (1) dynamic weight decision depending on the external events impact factors fusion for the linear combinations of two predictors,
- (2) general rules modification and prediction adjustment for the higher precision,
- (3) traffic rules application, such as left/right turn forbidden and speed limit, and
- (4) default rule actions for missing values.

#### 3.1. TTP function

As mentioned above, travel time along a selected path can be estimated by summing up the links travel time and intersection delays, as shown in Eq. (1). In the road network of an urban area, an ( $O, D$ ) pair may have many path choices, and each path may consist of several road links. As we know, there are many strategies for choosing the candidate paths, such as shortest path first, expressway first strategies, etc. In this paper, we adopt the top  $k$  used paths selection heuristic, i.e., each candidate path will be evaluated by our model to estimate the travel time, and the top  $k$  candidate paths from the historical database with the 1st to  $k$ th lowest evaluated travel time will be suggested. Once the candidate paths have been decided, travel time along each candidate path can be predicted by summing up the travel time of the links in the path and the intersection delays between consecutive links. Assume  $P_j(O, D)$  is a link set in the candidate path  $j$ , each  $L_i$  is the  $i$ th link from the origin in  $P_j(O, D)$ . The estimated travel time for path  $j$  can be formulated as Eq. (3), where  $T_c$  and  $T_h$  are the two predictors for real-time and historical prediction, respectively;  $D_{L_i, L_{i+1}}$  stands for the intersection delay between two adjacent link  $L_i$  and  $L_{i+1}$ , and  $\alpha_i, \beta_i$  are the weight control variables of the link  $L_i$ . As the travel time of each candidate path has been estimated, the travel time of a given ( $O, D$ ) pair can be estimated by choosing the minimum

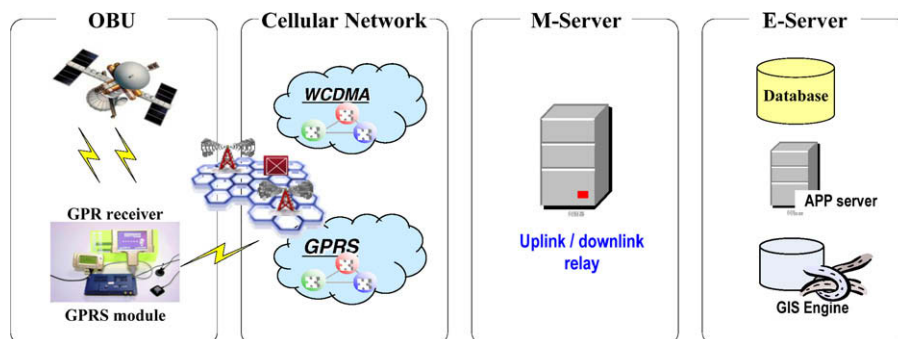


Fig. 1. Components of LBS system.

estimated travel time from the 1st candidate to the  $k$ th candidate path, as depicted in Eq. (4), and thus become the suggested path.

$$T_j(O, D, t) = \sum_{\forall L_i \in P_j(O, D)} (\alpha_i \cdot T_c(L_i) + \beta_i \cdot T_h(L_i, t)) + \sum_{\forall L_i \in P_j(O, D)} D_{L_i, L_{i+1}} \quad (3)$$

where  $\alpha_i + \beta_i = 1 \quad \forall i$

$$T(O, D, t) = \text{Min}_{j \in \{1..k\}} \{T_j(O, D, t)\} \quad (4)$$

### 3.2. System architecture

As shown in Fig. 2, the architecture of the proposed knowledge based TTP model includes four phases: traffic information generation, traffic patterns mining, rules construction, and travel time prediction. Besides, LBS-based applications, road network database in GIS engine and external traffic events data sources are the major inputs for the TTP model. There are two information flows in this architecture: real-time information flow drawn by solid line, sending real-time traffic information direct from Phase I to Phase IV, and batch running historical information flow drawn by dotted line. In the batch process flow, each phase outputs the result for the next phase. For example, journey set generated from Phase I will be fed into Phase II as input, and traffic patterns mined from traffic patterns mining module are fed into the pattern to rule transformation module in Phase III. In Phase IV, external events as well as real-time traffic information are read into inference engine as facts which fire the related rules to activate travel time inference process.

### 3.3. Ontology for the TTP system

As shown in Fig. 3, the TTP ontology designed by cooperation of the traffic domain expert, knowledge acquisition engineer, and system design engineer to organize the TTP inference structure consists of several design concepts and the relationships among these concepts, where one type of concepts is meta-concept drawn by red circle encrypting the domain knowledge donated by the traffic domain expert, or process control knowledge designed by system engineer, and the other is normal concept drawn in black circle representing the static knowledge or mining knowledge generated in the rules construction phase.

There are several relationships between these concepts, such as “is-a”, “includes”, “modify” and “related-to”. These relationships

connecting concepts in the ontology represent some interactions among them. For example, “real-time TTP predictor” concept includes a “Path selection” concept, and “Path selection” concept includes “Path to link decomposition” concept. When the TTP predictor concept is started, path selection process is then fired to find out the candidate paths between origin and destination, and the path selection process then fires the path to link decomposition process to get the links and intersections to be used to evaluate the travel time in each candidate path. In the following sections, some major concepts in the TTP are detailedly discussed to outline this TTP ontology workflow.

#### 3.3.1. Heuristic path selection

Path selection module is activated by the travel time predictor modules by the “includes” relationship in the TTP ontology. Path selection problem in the urban road network is much more complicated than that in freeway since only few paths can be chosen in freeway path routing for a give (O,D) pair. To minimize the path travel time, many strategies had been proposed for a routing path selection based on a given (O,D) pair in the urban network, such as shortest path first, expressway first or signal less path first. The common goal in these strategies is to minimize the path traveling time. To cope with the path selection problem, heuristic and domain knowledge are used in this paper. Since most of the taxi drivers can most likely select the heuristic optimal path according to their experience and the current traffic status. Our idea is to select top  $k$  paths from the journey set in the historical database as the candidate paths according to the request (O,D) pair and journey start time  $t$ .

#### 3.3.2. Historical predictor

Historical predictor and real-time predictor inherited from the TTP concept include two parts: link travel time estimation and intersections delay prediction. After historical predictor inferring the travel time of each link and intersection delays in the candidate path, the total amount of the links travel time and intersection delays is the predict result. Link travel time estimation is inferred by rules obtained from the spatiotemporal congestion patterns. Intersection delay prediction is reasoned by the rules obtained from the intersection delay patterns, which are classified by through delay (TD), right turn delay (RTD) and left turn delay (LTD). The rules transformed from these patterns are stored into the knowledge

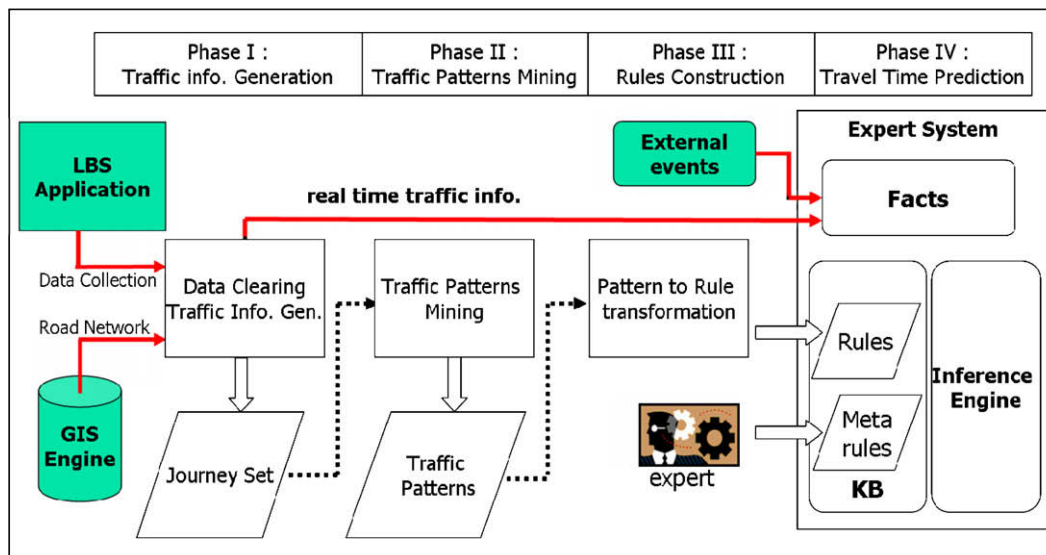


Fig. 2. Architecture of rule based real-time TTP system.

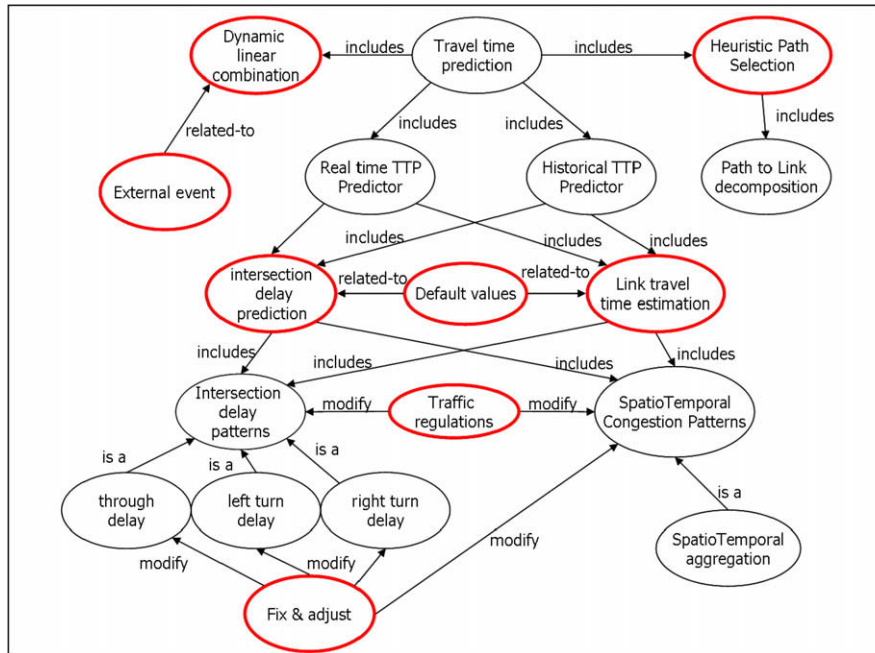


Fig. 3. Design ontology of the TTP system.

base (KB). When  $(O, D)$  pair request as well as journey start time are fed into the expert system as facts, the inference engine automatically fires the rules related to the historical TTP prediction. Meta-concept components in this ontology help to fill out the missing values and fixed some outliers. For example, if one rule shows that one link can be traveled at a speed in peak hour which is much higher than the speed limit, the meta-rule in “fix & adjust” component might fix and replace it with the speed limit of that link.

### 3.3.3. Real-time predictor

Real-time TTP of a candidate path can be done by summing up the travel time of each links and intersections delay which constitute the path, where the current travel time of a link can be easily done by dividing the link length with current average traveling speed of that link, and the real-time intersection delay is obtained from the real-time traffic information generated in Phase I. In the case of missing current speed of some links, a speed evaluation meta-rule given by domain expert in the “default values” concept is fired to give a default speed depending on their heuristic experience and spatiotemporal conditions. For example, domain expert may give default speed of midnight on non-holiday as 20% more than the speed limit of the link. The process of determine the missing value in intersection delay is done in the similar way.

### 3.3.4. Dynamic linear combination

“Dynamic linear combination” concept incorporates with “external event” concept to provide the real-time event response ability for the TTP model, which raises the precision of TTP and makes it more practicable. Real-time event response mechanism consists of several meta-rules, which is designed to handle the external events by dynamic tuning the weight of two predictors:  $T_c$ ,  $T_h$  through weight control variables:  $\alpha$  and  $\beta$ . For example, if the system receives a current external event, such as car accident on a link in the candidate path, event handling meta-rules will then reduce the weight of historical predictor ( $T_h$ ) and raise the weight of real-time predictor ( $T_c$ ). Because the effect of that car accident will be reflected at the corresponding link immediately, raise the weight of real-time predictor might get higher precision. On the other hand, some meta-rules may raise the weight of historical predictor

if the following two conditions are satisfied: if there is no current event, and the support and confidence of the related patterns are higher than the threshold set by the expert. It means that there is a strong support that traffic status most likely regresses to the intents of related historical patterns. Therefore, raise the weight of historical predictor might get higher precision.

### 3.4. Phase I: traffic information generation

Phase I includes processes of data collection, preprocessing, and transformation from LBS applications. The raw data collection collects the uplink and downlink interaction logs between OBU and LBS backend system. Each collected record is transformed to TIS as Eq. (2) depicted, including location, time, speed, direction, status of a vehicle at somewhere the OBU uplinks packet to the backend. The data collection, filtering and traffic information generation processes in this phase can generate traffic information in real time and then store it into to real-time road network status table for the expert system real-time TTP inference.

### 3.5. Phase II: traffic patterns mining

The generated traffic information is also stored in the historical database for traffic patterns analyzing. Traffic information generation module combines the continuous TISs of the same vehicle having explicit starting point and destination, and save them to a journey set table. The following four traffic patterns can be mined from historical journey set (Tsai et al., 2005): spatial and temporal aggregation (STA), through delay (TD), left-turn delay (LTD), and right-turn delay (RTD). The STA pattern is used for the link travel time estimation and the remaining three patterns are used for predicting intersection delays. The STA patterns are mined from historical traffic database by aggregating the TISs by spatial, temporal, and event dimensions, classifying the congestion level of the link by the attributes and traveling speed of that link, and calculating the support and confidence for each STA pattern. Spatial dimension stands for the link identification attribute in the urban road network, and temporal dimension is the classified indices of time domain, which includes: peak or off-peak hour, holiday or

workday, etc. Congestion level is determined by the ratio of average speed and speed limit of the link in the same spatial and temporal condition. For example, average speed of 40 km/h in the off-peak hour in workday is classified by free flow state in street, but will be classified by strongly congestion in freeway. Support and confidence can be calculated by aggregating the TISs at the same spatiotemporal conditions. The format of STA pattern is listed in Eq. (5):  $S_{id}$ , stands for link id,  $T_{id}$  is temporal id,  $E_{id}$  means the event condition of the pattern, such as normal, car accident, and road construction.  $C_g$  stands for congestion level, which is normalized by the link attribute (speed limit and link type) of target link ( $S_{id}$ );  $Sup$  and  $Con$  stand for support and confidence of the STA pattern, respectively. An example of STA pattern ('L1', 'W, N', 'N', 9, 0.3%, 65%) means that the link [L1] is in free flow state (congestion level 9) at non-peak hours of workday and the confidence of this pattern is 65%, support is 0.3%.

$$[STA] : (S_{id}, T_{id}, E_{id}, C_g, Sup, Con) \quad (5)$$

Intersection delay is the delay between two consecutive links, which is mostly caused by signal delay and queuing delay. Estimation of the intersection delay is too complex to be defined by mathematical model. Here we proposed two methods to estimate the intersection delay: traffic patterns and expertise. The expertise method defines the default intersection delay (TD/LTD/RTD) for each intersection, which will be a substitute when missing value or outlier condition is happened in traffic patterns method. Traffic patterns related to intersection delay can be classified by TD, LTD and RTD patterns, which are the delays of three possible directions from one link connects to another link, and can be mined from the historical journey set. Eq. (6) shows the general format of intersection delays (TD/LTD/RTD), where  $P$  is the pattern type,  $SO_{id}$  and  $SI_{id}$  are the two consecutive links ID where vehicle leaves out the link  $SO_{id}$  and comes into the link  $SI_{id}$ ,  $T_{id}$  is the temporal id,  $D_{avg}$  is the average delay time of this intersection, and  $Sup$ ,  $Con$  are the support and confidence of the pattern, respectively. For example, ('RTD', 'L1', 'L2', 'W, P', 40, 0.2%, 75%) represents that in the peak hours of workday, it takes 40 s to do a right turn from link 'L1' to link 'L2', and the support is 0.2%, confidence is 75%.

$$[TD/LTD/RTD] : (P, SO_{id}, SI_{id}, T_{id}, D_{avg}, Sup, Con) \quad (6)$$

Intersection delay patterns can be discovered by sequential pattern mining on historical traffic database by spatial and temporal sequences. Each sample of intersection delay must be in a journey which contains two consecutive TISs with different links. Fig. 4

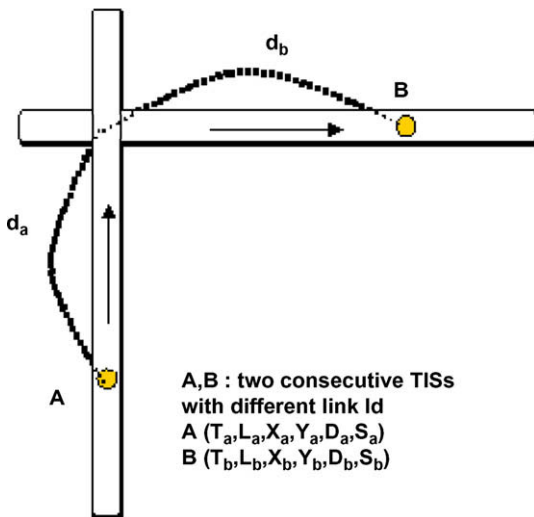


Fig. 4. Intersection delay example: RTD.

shows an example of RTD pattern: a probing vehicle driving north and then turn right to east, it reports TIS at location A of Link  $L_a$  and consecutively reports TIS at location B of Link  $L_b$ . The symbols of the TIS format ( $T, L, X, Y, D, V$ ) in Fig. 4 stand for timestamp ( $T$ ), link id ( $L$ ), coordinates ( $X, Y$ ), direction ( $D$ ) and speed ( $V$ ), respectively. The distance  $d_a$ ,  $d_b$  in Fig. 4 stands for the distance from A, B to the intersection of links  $L_a$  and  $L_b$ , respectively. Assuming that in the short period time interval between  $T_b$  and  $T_a$ , the vehicle is driving at the speed of  $V_a$  at link  $L_a$  and  $V_b$  at the link  $L_b$ . Then the right turn delay (RTD) time from the  $L_a$  to  $L_b$  can be estimated by subtracting travel time of  $d_a$  and  $d_b$  from elapsed time between two TISs ( $T_b - T_a$ ).

In the case of missing value or low confidence in some intersections, intersection delay can be easily estimated by another strategy: expert heuristics. Average delay time in the action part of TD/LTD/RTD rules for each consecutive links are provided by human experts and stored in the knowledge base. This strategy has two advantages: expert heuristics can be a stand alone prediction method of intersection delay, or it can be a supplement of intersection delay pattern when there are missing values in some intersections.

### 3.6. Phase III: rules construction

Two kinds of rules are constructed in this phase: rules transformed from the traffic patterns, and meta-rules donated by domain experts. The traffic patterns, mining from the historical journey set in previous phase, will then be transformed to the format of if-then rules by mapping the condition and action parts of the rule from attributes of the patterns, and decide the support and confidence of each generated rule by computing the probabilities of the rule. The if-then format rules are stored in the knowledge base, and can be easily read by inference engine and then make TTP inference. For example, STA patterns of Eq. (5) can be transformed to if-then rules by combining the link attribute in the traffic network database, as list in Eq. (7). The  $S_{id}$ ,  $T_{id}$ ,  $E_{id}$  information are transformed to the condition part of the if-then rule, and the congestion level ( $C_g$ ) can be transformed to estimated speed in the action part by considering the link attributes (speed limit and category). And thus the estimated travel time can be calculated by dividing the length of the link with the estimated speed. Eq. (8) shows the link travel time (LTT) estimation rule, which cooperates with STA rule and link attributes to compute the link travel time.

$$[STA \text{ rule}] \quad (7)$$

```

IF
  ( $S_{id}$  and  $T_{id}$  and  $itE_{id}$ )
THEN
  Congestion level at the link( $S_{id}$ ) =  $C_g$ ,
  Support =  $Sup$ ,
  Confidence =  $Con$ 

```

$$[LTT \text{ rule}] \quad (8)$$

```

IF
  ( $C_g = L$ ) and  $Cate(S_{id})$ 
THEN
  Speed at Link( $S_{id}$ ) =  $S$ ,
  LTT at Link( $S_{id}$ ) =  $Length(S_{id})/S$ 

```

### 3.7. Phase IV: travel time prediction

Expert system technology is used in this phase for the actual TTP job. There are three inputs of the expert system:  $O, D$  pair as well as start off time ( $t$ ) are the user's request coming from the user interface, and external events module gets the real-time traffic

events from the external data source: e-IOT (e-IOT) real-time traffic events database. These inputs as well as current traffic status coming from Phase I are loaded to the expert system as facts. The rules generated from the traffic patterns in Phase III and the meta-rule donated by domain experts are loaded into the knowledge base.

#### 4. System implementation

The TTP prototype system was implemented based on a real-time LBS-based application: taxi dispatch system (TDS) (Liu et al., 2004). The TDS is an online 7 \* 24 system operated in Taipei urban area (as shown in Fig. 5), and the current fleet size is about 500 taxis, where the OBU reports its current status based on following events: periodically (30 s), spatial trigger event, or some business events, such as dispatch/response event and customer on/off taxi events. The raw data in TDS system had half a million records per day, which was a good data source for this prototype TTP system. At the data collecting and clearing phase, the OBU report raw data has been collected and translated to TISs in a period of 5 min in order to catch the real-time traffic information, and all the TISs are filtered out except the OBU in 'dispatch' or 'occupied' state since the traffic information extracted from these two states is meaningful. Besides, the OBU report will be filtered out if the location of its TIS is not in the interested links (links in Fig. 5). Historical traffic information consists of journey sets by combining the raw data and the GIS road network. A journey is a tour consisting of a set of sequence TISs between origin and destination. The 'dispatch' state journey starts from the dispatch location to the customer's location, and 'occupied' state journey starts from the customer's location to customer's destination.

As shown in Fig. 5, the target area of this prototype system is focused on the arterials in Taipei urban area, and each arterial may have one or several links. Link attributes including link category, length, direction, speed limit, average signal delays and geographical coordinate vectors are defined and default values are given by domain experts to facilitate link travel time estimation. At the traffic patterns mining phase (phase II), traffic level (1–9) is classified by aggregating the TISs at temporal and spatial dimension and normalized by two link attributes: category, speed limit. For example, traffic level 9 means the link is in free flow state and the traveling speed is near around the speed limit; on the other hand, traffic level 1 represents that the link is in extremely congestion status.

The TTP prototype system was implemented by the expert system shell, (DRAMA, 2003), a new object-oriented rule model (NORM) (Lin, Tseng, & Tsai, 2003) knowledge modeled rule base system platform implemented using pure Java language, includes Drama Server, Console, Knowledge Extractor, and Rule Editor. TTP rules generated from the Phase III and meta-rules donated by traffic experts are categorized to several knowledge classes (KC), and stored at knowledge base of DRAMA server. A real-time traffic information database: e-IOT is connected as an external data source. E-IOT is a centralized real-time traffic information database provided by institute of traffic of Taiwan government, which provides various traffic event information, including traffic block, traffic jam, construction, signal break, disaster, and accident. The external traffic information is used as a trigger input for weight combination knowledge classes (meta-rules), which dynamically tunes the weighted combination of current and historical travel time predictors. The essence of the weight tuning meta-rules can be induced in two principles: meta-rules raise the ratio ( $\alpha$ ) of current predictor when a current event is happening at some links, and return to the origin ratio when the event has been relieved.

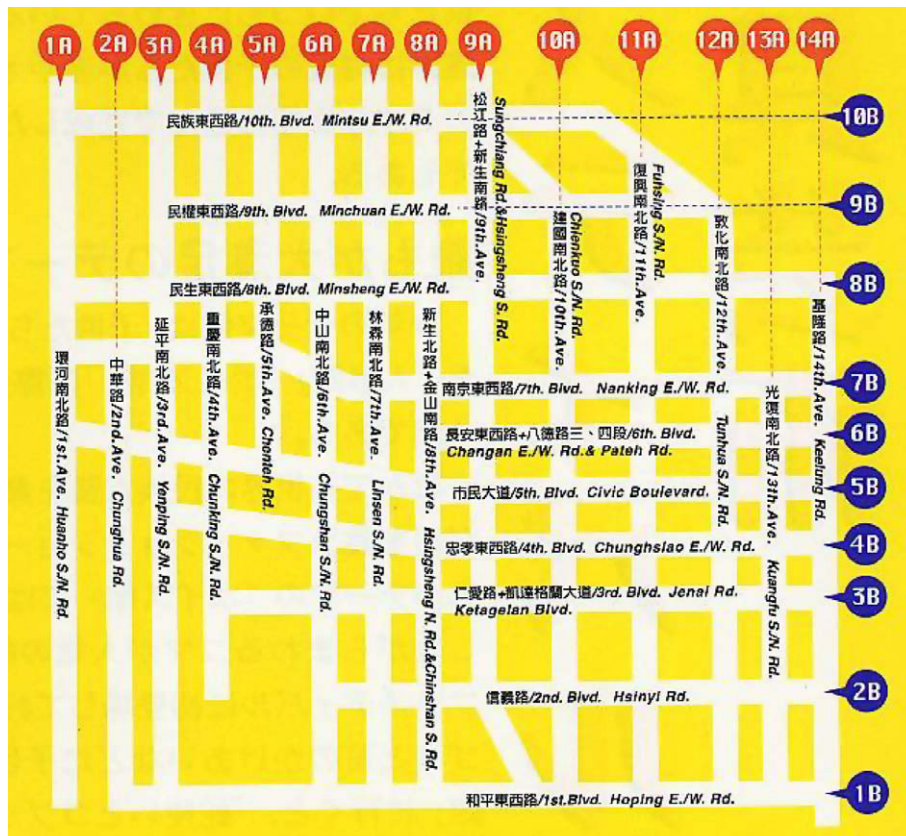


Fig. 5. Road network in Taipei urban area.

The second principle is that the weight of historical predictor ( $\beta$ ) will be raised when the support and confidence of related patterns are higher than the threshold set by domain experts.

#### 4.1. Experiment results

We collect five months (2006/02–2006/06) LBS raw data for the experiment, the data of first four months is for mining the traffic patterns and the fifth is for testing the TTP. In the experiment, three predictors: current-time predictor, historical traffic pattern predictor and weighted combination predictor had been implemented and compared. Current-time predictor makes TTP by summing up the travel time and intersection delays based on current speed on the links of the candidate path and related intersection delays. Historical traffic pattern predictor predicts travel time by reasoning the historical traffic patterns. The weight combination predictor predicts travel time by weighted combination of these two predictors based on the external traffic events and meta-rules. Two performance indices: relative mean errors (RME) and root mean squared errors (RMSE) are applied for comparing these predictors and listed as Eqs. (9), (10) where  $n$  is the number of prediction,  $X_i$  and  $\bar{X}_i$  present the travel time and prediction time, respectively. The last of linear combination predictor use the meta-rules to dynamically adjust  $\alpha$  and  $\beta$  variables with real-time events consideration, as mentioned before.

$$\text{RME} = \frac{1}{n} \sum_{i=1}^n \left| \frac{X_i - \bar{X}_i}{X_i} \right| \quad (9)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left| \frac{X_i - \bar{X}_i}{X_i} \right|^2} \quad (10)$$

Table 1 shows the RME and RMSE results of different predictors on workday (Monday–Friday). In this experiment, we randomly choose different  $O, D$  pairs at two peak hour sections: 7:30–9:00AM and 6:00–7:30PM and then calculate the RME and RMSE with each predictor. The result shows that current-time predictor has better precision than historical predictor. By combining domain expert knowledge in meta-rules, weight combination predictor has better performance than other two predictors in both RME and RMSE.

In the above experiment, the intersection delay is regarded as fixed variable so that all the three predictors use the same ID patterns for calculating intersection delays. In order to compare the ID patterns and human expertise, the experiment with the same data set has been made again by replacing ID patterns with human expertise. That is, human expert gives a default value for average turning delays instead of mining ID patterns from historical traffic database. The result in Table 2 shows that precision of using human expertise is not as good as ID patterns. This might be

**Table 1**  
TTP precision of three predictors (by ID patterns)

Predictor	Historical predictor	Current-time predictor	Weight combination predictor
RME (%)	20.5	13	10.8
RMSE (%)	24.82	18.13	15.92

**Table 2**  
TTP precision of three predictors (by expertise in ID)

predictor	Historical predictor	Current-time predictor	Weight combination predictor
RME (%)	29.6	21	20
RMSE (%)	35.78	32.21	30.55

improved by tuning the turning delay default value by domain experts.

## 5. Conclusion

Travel time prediction is useful and important for many ITS applications. Real-time travel time prediction for urban network is a complex task so that it is regarded as theoretically feasible but difficult to accomplish using traditional models. The proposed real-time knowledge based TTP model has demonstrated that TTP for urban network could be achieved cost-effectively by utilizing the raw data of LBS-based applications. Dynamic combination of real-time and historical predictors takes the advantage of the two predictors and promotes the TTP precision one step further. System implementation on urban network based on a deployed online TDS system and the results show that the precision of TTP can be achieved in tolerable range.

## Acknowledgements

This work was partially supported in part by National Science Council of the Republic of China under Grant No. NSC96-2752-E009-006-PAE. The authors wish to express their appreciations to the Research Institute of Chunghwa Telecom for kindly sharing the raw data of their online TDS system.

## References

- Bajwa, S. I., Chung, E., & Kuwahara, M. (2004). An adaptive travel time prediction model based on pattern matching. In *Proceedings of the 11th world congress on ITS*.
- Chien, S., & Kuchipudi, C. M. (2003). Dynamic travel time prediction with real-time and historic data. *Journal of Transportation Engineering*, 608–616.
- Chung, E. (2003). Classification of traffic pattern. In *Proceedings of the 11th world congress on ITS*.
- Chung, E., Sarvi, M., Murakami, Y., Horiguchi, R., & Kuwahara, M. (2003). Cleansing of probe car data to determine trip OD. In *Proceedings of the 21st ARRB and 11th REAAA conference, Cairns, Australia*.
- Chung, E. et al. (2003). Travel time prediction: Issues and benefits. In *Proceedings of the 10th world conference on transport research*.
- DRAMA (2003). Coretech Inc. <<http://www.ctknow.com.tw/rule.htm>>.
- e-IOT, Real-time traffic events database supported by the Institute of Transportation MOTC. <<http://e-iot.gov.tw/>>.
- Iroy, T., & Kuwahara, M. (2001). Simulation analyses of traffic congestion alleviation by demand spreading over time. *Journal of the Eastern Asia Society for Transportation Studies*, 4(4), 161–174.
- Jiang, G., & Zhang, R. (2003). Travel time prediction for urban arterial road. In *Proceedings of the IEEE international conference on intelligent transportation systems* (Vol. 2, pp. 1459–1462).
- Lee, W. H., Jeng, B. S., Tseng, S. S., & Wang, C. H. (2004). Electronic toll collection based on vehicle-positioning system techniques. In *Proceedings of the IEEE international conference on networking sensing and control* (pp. 643–648).
- Lin, H. E., & Zito, R. A review of travel-time prediction in transport and logistics. In *Proceedings of the Eastern Asia Society for transportation studies* (Vol. 5, pp. 1433–1448).
- Lin, Y. T., Tseng, S. S., & Tsai, C. F. (2003). Design and implement of new object-oriented rule base management system. *Expert Systems with Applications*, 25, 369–385.
- Lin, W. H., Kulkarni, A., & Mirchandani, P. (2004). Short-term arterial travel time prediction for advanced traveler information systems. *Journal of Intelligent Transportation Systems*, 8, 143–154.
- Liu, H. Y., Wang, C. H., Shieh, V. S., & Jeng, B. S. (2004). An intelligent taxi dispatching management system. In *Proceedings of cross strait conference on ITS* (pp. 111–117).
- Mark, C. D., Sadek, A. W., & Rizzo, D. (2004). Predicting experienced travel time with neural networks: A PARAMICS simulation study. In *Proceedings of the IEEE conference on intelligent transportation systems* (pp. 906–911).
- Nakata, T., & Takeuchi, J., (2004). Mining traffic data from probe-car system for travel time prediction. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, Seattle, WA, USA* (pp. 817–822).
- Rice, J., & van Zwet, E. (2004). A simple and effective method for predicting travel times on freeway. *IEEE Transactions on Intelligent Transportation System*, 5(3), 200–207.
- Takahashi, H., Takahashi, S., & Izumi, T. (2003). Travel time measurement using genetic algorithm in vehicle sequence matching processing. In *Proceedings of the 2nd ITS symposium, Osaka*.
- Tsai, S. H., Lee, W. H., & Tseng, S. S. (2005). A spatiotemporal traffic patterns mining on LBS-based application. In *Proceedings of the 10th TAAI conference (CD-ROM)*, Kaohsiung, Taiwan.



- Wang, C. W., Chiu, C. C., Jeng, S. D., Hsiao, S. R., Wei, L. G., Chao, C. H., et al. (2003). A Geocoding application on GIS using address data: Case study of Taiwan address database. In *Proceedings of the TGIS 2003 annual conference, Taipei, Taiwan* (pp. 25–26).
- Wen, Y. H., Lee, T. T., & Cho, H. J. (2005). Missing data treatment and data fusion toward travel time estimation for ATIS. *Journal of the Eastern Asia Society for Transportation Studies*, 6, 2546–2560.
- Wu, C. H., Ho, J. M., & Lee, D. T. (2004). Travel time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation System*, 5(4), 276–281.
- Yang, J. S. (2005). Travel time prediction using the GPS test vehicle and Kalman filtering techniques. In *American control conference, June 8–10, Portland, OR, USA* (pp. 2128–2133).
- Further reading**
- Kawahara, T., Kamijo, S., & Sakauchi, M. (2005). Travel time measuring by using vehicle sequence matching between adjacent intersections. In *Proceedings of the IEEE international conference on intelligent transportation systems* (pp. 712–717).
- Liu, H., Zulen, H., Lint, H., Chen Y., & Zhang, K. (2005). Prediction of urban travel times with intersection delays. In *Proceedings of the IEEE international conference on intelligent transportation systems* (pp. 1062–1068).
- Safa, I., Fernando, V. L., Richard, T. S., & Bongki, M. (2005). Spatiotemporal aggregate computation: A survey. *IEEE Transactions on KDE*, 17(2), 271–286.
- Sarvi, M., Horiguchi, R., Kuwahara, M., Shimizu, Y., Sato, A., & Sugisaki, Y. (2003). A methodology to identify traffic condition using intelligent probe vehicles. In *Proceedings of the 10th world congress on ITS, Madrid*.
- Wei-Shinn, K., Roger, Z., Haojun, W., & Chi-Ngai, W. Adaptive nearest neighbor queries in travel time networks. In *Proceedings of the ACM GIS* (pp. 210–229).
- Yoo, B. S., Kang, S. P., & Park C. H. (2005). Travel time estimation using mobile data. In *Proceedings of the Eastern Asia Society for transportation studies* (Vol. 5, pp. 1533–1547).
- Zhang, H. S., Zhang, Yi, Li, Z. H., & Hu, D. C. (2004). Spatial-temporal traffic data analysis based on global data management using MAS. *IEEE Transactions on ITS*, 5(4), 267–275.