Computing Journey Start Times with Recurrent Traffic Conditions

Rahul Kala and Kevin Warwick
School of Systems Engineering, University of Reading, Whiteknights,
Reading, Berkshire, United Kingdom
rkala001@gmail.com, k.warwick@reading.ac.uk
Tele: +44-7424752843

Citation: R. Kala, K. Warwick (2014) Computing Journey Start Times with Recurrent Traffic Conditions. IET Intelligent Transport Systems, 8(8): 681 – 687.

Final Version Available At: http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?arnumber=6955058

Abstract: In this paper we discuss how technology can be used to effectively solve the problem of deciding on journey start times for recurrent traffic conditions. The developed algorithm guides vehicles to travel on more reliable routes which are not easily prone to congestion or travel delays, ensures that the start time is as late as possible to avoid the traveller waiting too long at their destination and attempts to minimize the travel time. Experiments show that in order to be more certain of reaching their destination on time, a traveller has to leave early and correspondingly arrive early, resulting in a large waiting time. The application developed here asks the user to set this certainty factor as per the task in hand, and computes the best start time and route.

Keywords: start time prediction, traffic forecasting, travel time prediction, intelligent transportation system.

1. Introduction

The problem of start time prediction deals with deciding the time a person should leave from the source in order to reach their destination by a given time. The problem is commonly seen in everyday life. A person may have to meet some distant relatives, catch a train or flight, etc. We however do not necessarily include the problem of going to the office in this category as this task is performed everyday and the person has the scope to experiment with different start times and choose the best one. In the former the person is either unfamiliar with the route and the traffic trends on it, or is not updated about any changes in the traffic trend. In the office case the person tunes the start time every day.

In making a decision regarding the journey start time, only recurrent trends [1] tend to be considered. The start time decision needs to be made before the journey starts, when the information regarding non-recurrent trends is unavailable or is largely uncertain. That said, a transportation authority may occasionally advertise the possibility of slow traffic on some roads due to pre-planned reasons, in which case more spare journey time may be allowed for. In the case of arrival at a time which is later than the desired time there is a penalty which entirely depends upon the purpose of travel. For example in the case of catching a train or plane the person may actually miss their connection and may have to cancel their entire plan. However in the case of meeting friends and relatives the penalty of late arrival is possibly negligible.

The key contributions of this paper are: (i) We propose here decentralized agents at intersections which record traffic speeds and variance along with time. The use of centralized agents (or single agent systems) for such an approach is common, however this is not a scalable approach. The use of decentralized agents for traffic speed monitoring is also common. Here recording the extra variation factor helps in answering user queries. (ii) We study a new problem of start time prediction, where the users may adapt the algorithm based on the penalty of late arrival. A single factor governs the performance. Guidelines enable the user to set the parameter as per their requirements. (iii) Using the existent notion of advanced driver information systems, we simultaneously solve the twin problems of start time prediction and routing. (iv) We propose a graph search method to compute the route and start time for the vehicle. The algorithm attempts to select a route which is shortest in length, has high reliability and gives, as output, the latest possible journey start time.

2. Related Work

Although the problem, as described in section 1, has high relevance, it has not been appreciably studied in the literature in its direct form. The closest work is that of Kim et al. [2]. Here the authors addressed three issues, namely driver attendance time, vehicle departure time and routing policy using Markov decision process. Searching for the optimal policy in a time varying Markov decision process is a time consuming process and possible only for small maps.

Claes et al. [3] used a decentralized routing strategy where every vehicle considered all possible routes. For every route the vehicle queried for the total travel time based on the expected vehicle density. Weyns et al. [4] used traffic microsimulation to estimate both travel times and travel speeds. However the problem with these approaches is that all vehicles need to be intelligent. Further, as time progresses, more vehicles start their journeys which increases the expected density and hence the travel time. In the proposed work, the discovery of new vehicles starting up might mean that no route could subsequently be guaranteed for a vehicle to reach its destination on time.

Some algorithms have been designed specifically to solve the problem of traffic congestion [5] which is usually the source of excessive delays, for example by the use of digital pheromones [6-7]. Here the vehicles deposited pheromones on the routes they followed, which evaporated with time. Later vehicles avoided routes having a high pheromone content. Again a limitation of this method is that all the vehicles need to be intelligent.

Another related problem is that of travel time prediction. The problem is a challenging one due to the stochastic nature of travel time. van Hinsbergen et al. [8] used Bayesian Neural Networks. The historic de-noised signal representing the traffic flow was used by a neural network for prediction. Other neural network approaches include [9-12]. Uncertainties associated with the prediction become very high if the planning is being done too much in advance. Our approach exploits the recurrent nature of the traffic flow for learning.

Routing in real traffic sense is a stochastic and time varying problem. For the same reasons a standard shortest path graph search cannot be used. Miller-Hooks and Mahmassani [13] employed methods to compare two probabilistic paths and selected a path to be better only if it dominated in all probability realizations. The authors built a pareto front of non-dominant solutions. Similar work exists in [14]. The approaches are computationally very expensive and work only for small maps. A large amount of research has been carried out in hierarchical planning (e.g. [15-17]) where the researchers believe a shortest path search algorithm may itself not be able to search for an optimal path in very large cities.

In our work we have converted the problem into a deterministic equivalent, where the required certainty (as defined by the user) is used to get the best time estimate. This enables us to solve the problem within small computation times. The actual travel of the vehicle is stochastic which may not follow the timestamps computed by the algorithm, but the best trade off is sought between ensuring safety and running late as opposed to not reaching the goal too early. We consider the certainty of reaching the destination on time instead of the best expected utility, which better models the requirement.

3. Problem Statement

Assume that a person needs to travel via a transport network in order to reach a destination L at the latest by a scheduled time T, while the journey needs to start from a location S. The road network graph G of the city is assumed to be known. The problem is to compute the start time T_s and route R. Here R denotes a set of vertices $<S, V_1, V_2, \ldots L>$ starting from the source S and ending at the destination L. Let the duration of the journey be denoted by T_t and time of reaching the destination by T_f (which gives $T_s + T_t = T_f$). For a given T_s, the values T_t and T_f are stochastic in nature as different runs of the same vehicle may differ in travel durations and finish times due to the presence of other vehicles, traffic lights, etc. While T denotes the scheduled time that the vehicle aims to reach the destination, the vehicle may or may not be able to do so because of traffic uncertainties. The actual time at which the vehicle reaches the destination is denoted by T_f. Let $P(t \leq T | T_s, R)$ denote the probability that given the start time T_s and the route R, the vehicle reaches destination L at a time on or before the desired time T. Here $P(t \leq T | T_s, R)$ is a probability distribution while T_f and T_t are unit samples from related distributions.

The objectives of the algorithm are: (i) The start time must be as late as possible (*maximize* T_s). (ii) The route R must be the fastest way to reach the destination, or the travel time should be minimal (*minimize* T_t). (iii) The route R should be as reliable as possible (*maximize* $P(t \leq T | T_s, R)$). (iv) If for any reason the person reaches the destination before the scheduled time T, this time should be as little as possible (*if* $T_f \leq T$, *minimize* $T - T_f$, *high penalty otherwise*).

4. Algorithm

The algorithm uses intelligent agents to enable computation of the start time and the route. The agents record the mean travel speeds and the variance at the different times of day and on different days. This information is used by a graph search algorithm to compute the route and hence the start time. The general algorithm framework is given in Figure 1.

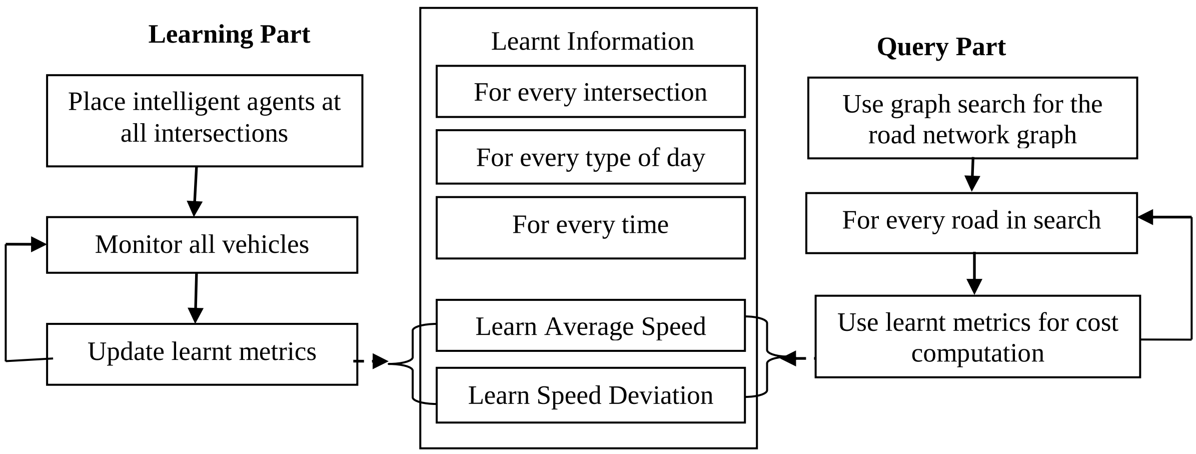

Figure 1: The general algorithm framework

4.1 Learning Travel Speeds

The assumption behind the algorithm is that the traffic is recurrent. Hence traffic flow and density observed at a particular time of the day would be similar to that observed at the same time on a similar day. *Similar days* means the days of the week when the traffic is expected to be similar. Traffic on Mondays and Sundays clearly has different trends while the trends may be only slightly different for Mondays and Tuesdays. However traffic in general shows a trend of a gradual increase along with time, seasonal variations, noise, etc. Hence an element of *learning* is introduced.

We assume that the road network graph G has a total of $|V|$ vertices where every vertex is an intersection. Each of these intersections is occupied by an intelligent agent. The agents monitor the vehicles and record the speed information (e.g. see [8]). Let $speed(V_1, V_2, t, d)$ denote the average speed in going from intersection V_1 to intersection V_2 at time t of the day and at a particular day-type d of the week. Here time t is broken down into buckets of 10 minutes. The assumption is that the average speeds in real life traffic do not change much in an interval of 10 minutes. Making the time interval too small results in too many parameters to learn, which may hence be difficult to compute and uncertain due to less data. Too large a time interval may show a high deviation of speeds within the time interval as any change in trend within the time interval cannot be captured.

Suppose a vehicle A left the intersection V_1 at time t_1 as observed by the agent at V_1 and is seen leaving the intersection V_2 at time t_2 by the agent at intersection V_2. The agents may be sophisticated to track and identify the vehicle [8] or intelligent vehicles [18-19] may themselves communicate their identity. The agent at intersection V_2 hence observes the average speed given by (1).

$$speed(A) = \frac{\| V_2 - V_1 \|}{t_2 - t_1} \tag{1}$$

Here $\|V_2 - V_1\|$ denotes the distance between intersections V_1 and V_2. The average speed includes any time spent in waiting for traffic lights at V_2 (if any). In a related publication [20] we showed that vehicles may be waiting for prolonged times at the traffic crossings and hence accounting for this waiting time is important.

Learning the factor $speed(V_1, V_2, t, d)$ by the agent at V_2 is done using (2). This equation constantly adapts the speed to the changing traffic trends. The updated speed estimate $speed(V_1, V_2, t_1, d)^{new}$ is taken partly from the actual speed of the vehicle A and partly from the old speed estimate $speed(V_1, V_2, t_1, d)^{old}$. The old speed estimate may be initialized based on observations of a few initial vehicles or to the road's speed limit. Algorithmically the value of $speed(V_1, V_2, t_1, d)$ is constantly changed in consideration of the newly recorded speed of the vehicle A. As more and more vehicles pass by, their recorded speeds are used to correct the overall speed estimate. Only a fraction of the estimate, equalling learning rate, is taken from the vehicle *A* to eliminate any noise or slow driving preference of the vehicle *A*.

$$speed(V_1, V_2, t_1, d)^{new} = (1 - lr) speed(V_1, V_2, t_1, d)^{old} + lr.speed(A) \tag{2}$$

Here lr ($0 < lr \leq 1$) is the learning rate. A small value of learning rate implies that the algorithm is passive and does not capture any rapidly changing trends. A high value meanwhile denotes that the agent may treat any delay which is due to the personal preferences of the driver, or similarly small delays, as a change in trend.

The agent further measures the standard deviation $\sigma(V_1, V_2, t, d)$, given by (3). While the speed is learnt using a learning rate for each vehicle as it passes by, the deviation is measured for all the vehicles that passed by in the previous δ similar days. Here N denotes the number of vehicles considered for computing the deviation. Too small a value of δ might mean too few vehicles are considered for the computation, which in turn would mean uncertainty in the recorded deviation. Taking too high a value however might cause undue effects of historical data which may have changed with time. For most high density roads, a small value would suffice.

$$\sigma(V_1, V_2, t_1, d) = \sqrt{\frac{\sum_{\text{previous similar } \delta \text{ days}} \left(speed(V_1, V_2, t_1, d)^{new} - speed(V_1, V_2, t_1, d)^{old}\right)^2}{N}} \tag{3}$$

One of the key aspects is that we propose a *decentralized architecture*. In a practical system, with centralized approach there would be a large number of users attempting to compute the start time and route. They would all query the central server and occupy it for a long time. It may not be possible to simultaneously handle so many users. The centralized approach however makes the algorithm require a single connection, and hence the speed of the algorithm may be faster for the case of a single user.

In a decentralized approach the computation is spread across the agents. The graph search algorithm considers only competing routes, intersections corresponding to which are queried. An algorithm hence queries a small number of agents. Every agent has limited demands despite the total number of queries throughout the system being high. However this forces the search algorithm to make a large number of connections.

On many occasions *non-recurrent trends* appear in traffic in which the traffic flow is temporarily altered from the expected trends, and such an alteration will most likely not be seen in the future. Such trends must be identified and neglected. Hence if the assessed speeds are very different from the expected averages, learning is not carried out for that vehicle considering it as a non-recurrent trend. If such irregular trends continue in the future, the learning framework interprets it as some new trend and the learning continues. Many times such trends may be known apriori for example a football match, public event, etc. In such cases the transportation authorities may ask the algorithm to neglect such cases by pausing the learning.

4.2 Routing

The problem is to enable a vehicle to decide its starting time and route of travel. The expectation is to have the vehicle at the destination L at the pre-decided scheduled time T. Let $T(V_i)$ denote the latest time by which the vehicle must be at the intersection V_i so that it can hope to reach the destination L at a time T with a high probability. The algorithm proceeds by computing $T(V_i)$ for all the nodes. The value of $T(V_i)$ at the source is hence the starting time.

The problem is modelled as a graph search which goes from the goal towards the source, which is an inverted version of a regular graph search problem. In a regular graph search, starting from the source at a time 0, the intent is to reach the goal with the shortest time (or any other metric). In this problem it is known that $T(L)=T$, or that the vehicle should be at the destination at the latest by time T, while the same needs to be computed for the other nodes, especially the source.

The objective of the problem was to simultaneously maximize the start time T_s, minimize the travel time T_t, maximize the probability of reaching before the pre-determined time $P(t \leq T \mid T_s, R)$ and minimize the waiting time as much as possible (if any) or *if* $T_f < T$, *minimize* $T - T_f$. These objectives need to be fused into a single objective that the graph search might optimize. For a deterministic approach the objective to minimize the travel time T_t and minimize the delay are the same as the objective to maximize the start time T_s. However the factor $P(t \leq T \mid T_s, R)$ is contrary to all of the above. To be assured of reaching the destination on or before the desired time, one has to keep a large spare time which decreases the start time. A related problem is that the time variables mentioned are stochastic in nature which pressurizes the need to work with probability distributions, which is computationally expensive.

For these reasons the approach followed in this work is to *fix the probability* $P(t \leq T \mid T_s, R)$ which would normally be as required by the user, and *go forth with maximizing* T_s. Considering contradictory objectives, there has to be some way of making a trade-off between them. Considering the nature of the problem, only a user can decide whether he/she is travelling for a task where being late is not allowed or otherwise. This further converts the problem into a deterministic equivalent where a probability distribution may be replaced by the best value of the metric which lies above the performance threshold.

The graph search takes place from the destination with $T(L)=T$ and proceeds in pursuit of the source. Consider expansion of an intersection V_1. The node V_1 is said to be connected to all nodes V_2 such that $\langle V_2, V_1 \rangle$ is an edge. The vehicle needs to be at V_1 on or before $T(V_1)$ to reach the destination on time. The time taken to travel from V_2 to V_1, denoted by $Tm(V_1,V_2)$, may be given by (4). $T(V_2)$ may hence be given by (5).

$$Tm(V_1,V_2) = \| V_1 - V_2 \| / \left(speed(V_2,V_1,T(V_1),d) - \alpha.\sigma(V_2,V_1,T(V_1),d) \right) \tag{4}$$

$$T(V_2) = T(V_1) - Tm(V_1,V_2) \tag{5}$$

The expected speed of travel from V_2 to V_1 is $speed(V_2, V_1, T(V_1), d)$. However the actual speed may be different from the general expectation and hence a penalty is added which is proportional to the learnt deviation. A low deviation means that all vehicles on the road travel with almost the same speed and hence the learnt speed is reliable, while a high deviation means that the learnt speed is not reliable and hence an extra safety time has to be included. The factor α establishes a trade-off between the two opposing factors of the probability of reaching the destination on time and the latest arrival, and is set by the user in understanding his/her requirements.

It is possible at times that some roads, maybe for some parts of the day, are rather under used. Hence there may not be enough data with which to learn the speed on a road and the recorded deviation (σ) may therefore itself be unreliable. High deviation implies high unreliability and such roads are seldom used by drivers. Such roads have a high penalty and are hence unlikely to be used by the algorithm. Hence high deviation due to lack of data produces very limited undesirable effects. However lower deviation values due to lack of data indicate reliable roads which is not really the case. The algorithm may choose such a road due to its low penalty, while the vehicle may actually take longer.

Consider the case when a road is unused and only a single vehicle passed in a particular time frame, based on which the variance computations were made. Consider that this vehicle was driving fast and further did not have to wait for any traffic signal. However, now taking this road might not be that attractive. The problem is not having enough data for learning. Hence a minimum threshold σ_{min} of deviation has been set. The effective deviation used in the calculations would hence be the larger of σ_{min} and the learnt σ. σ_{min} ensures that no road is regarded as too reliable (or possesses a small deviation) which may actually be due to the small amount of data. A higher value of σ_{min} is too pessimistic an approach, wherein the algorithm does not trust the recorded deviations and prefers to take large margins on all the roads. A small value on the other hand has the risk of the road being selected despite an uncertain value due to a small amount of data.

The vehicle is projected to be arriving at V_2 at a time $T(V_2)$. In reality it may arrive a lot earlier. The actual speed and the speed at the projected time can hence be very different. The risk is whether the vehicle would still be able to reach V_1 on or before $T(V_1)$. Consider, as per the A* algorithm, a vehicle is expected to reach a particular node at 10:00 AM and the next node using the available road at 10:10 AM. These computations kept some slack time at every node. So it is possible that the vehicle reaches the node by 9:30 AM instead of the scheduled 10:00 AM. It is also possible that traffic at 9:30 AM is highly congested due to office crowding, which clears at 10:00 AM. Due to this the expectation at the node by the A* algorithm was a clear road, however since the vehicle reached the node early it saw congested roads. We need to prove that it can still reach the subsequent node at the latest by 10:10 AM. Otherwise the algorithm does not hold.

Suppose two vehicles *A* and *B* reach V_2 at times t_{2A} and t_{2B}. Suppose they reach V_1 at times t_{1A} and t_{1B}. If vehicle *A* reaches V_2 earlier, that is $t_{2A}<t_{2B}$, it can be ascertained that it also reaches V_1 earlier, that is $t_{1A}\leq t_{1B}$, provided that they travel by the same preferred speed and no single lane is reserved for high speed traffic which one of the vehicles can utilize while the other cannot. In a single lane case there is no way that *B* would come from behind and overtake *A*. In a multi-lane case, the vehicles arrange themselves so that the speed on all the lanes becomes equal and hence there is no way that *B* would come from behind and overtake a vehicle nearly moving parallel to *A*. In case the lanes are with different speed limits or the road is not densely occupied, the speeds of the lanes may be different. If *A* is running lane, it would certainly change to a high speed lane and the proof for a single lane case holds. Hence if the vehicle reaches V_2 at $t_{2A}<T(V_2)$ it should be able to reach V_1 at $t_{1A}\leq T(V_1)$.

The approach taken here uses the A* algorithm [21] from the goal, at each instance expanding nodes as per the priority queue. Every node V_2 with parent V_1 is associated with time $T(V_2)$ given by (5), time to destination $Td(V_2)$ which is the expected journey time from V_2 to the destination given by (6), heuristic time to source or $Ts(V_2)$ given by (7) and the total expected duration of the journey $Tj(V_2)$ given by (8). *velmax* is the maximum preferred speed of the vehicle. The nodes are sorted as per *Tj* values in the priority queue and the smallest value is taken for expansion.

$$Td(V_2) = Td(V_1) + Tm(V_1,V_2) \tag{6}$$

$$Ts(V_2) = \| V_2 - S \| / velmax \tag{7}$$

$$Tj(V_2) = Td(V_2) + Ts(V_2) \tag{8}$$

Once the A* algorithm terminates at the source. $T(S)$ is the start time and the parent information from S to L is used to compute the route to take. $Td(S)$ is the expected journey time.

4.3 Probability of reaching the destination on time

In section 4.2 it was stated that α controls the trade-off between the probability of reaching the destination on time and maximizing the start time. The factor however cannot be used to compute the precise probability. In fact computing the probability distribution over time is a rather hard and computationally expensive task. It is proposed that the relation of α to the probability can be conveniently studied by experimentation for every region. Considering that the duration of the journey does not vary alarmingly, the generality of the conversion is high. Such a study can be performed for different regions. The study can be presented to the user, who may then be able to fix a value of α depending upon the purpose of the journey.

5. Results

The approach was tested using reasonably realistic simulations. For the experiments the map of Reading, United Kingdom (Figure 2) was used which was obtained from Openstreetmap [22]. The isolated nodes were eliminated using a Depth First Search. The map had a total of 7765 road nodes. The speed limit was 40 miles/hour. The motion of the vehicles was made using the Intelligent Driver Model [23].

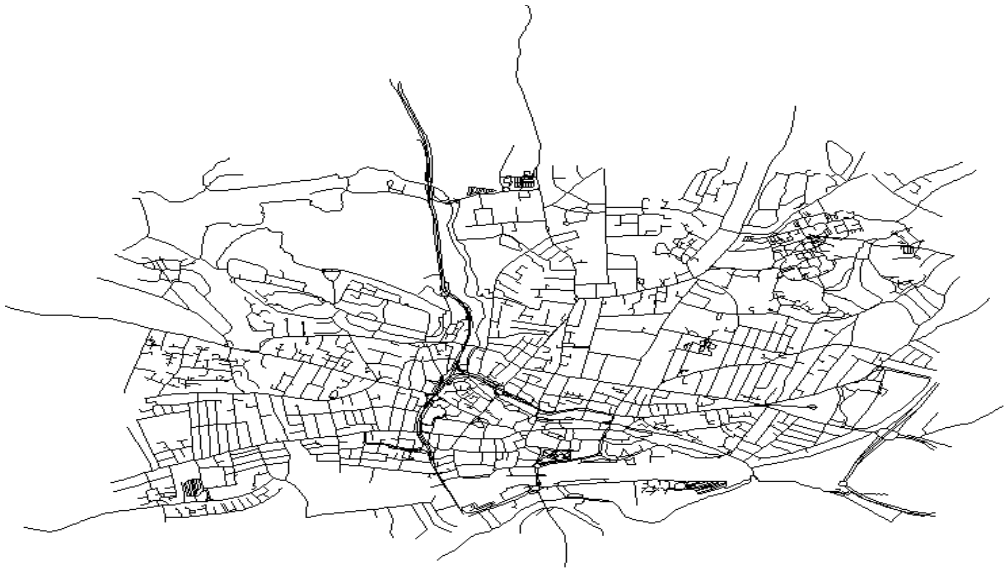

Figure 2: Map of Reading, UK used for experimentation

For all vehicles the origin was selected using a Gaussian distribution with mean centered outside the map's central point by a magnitude of half the radius. The angle of origin to the map's center (θ) was chosen randomly. The destination was also chosen from a Gaussian distribution with mean at half the map's radius. The angle of destination to the map's centre was chosen from a Gaussian distribution with mean located at $\pi + \theta$.

The first task was to learn the travel speeds for the recurrent vehicles. A pool of vehicles was generated. The generation of vehicles was done for a total of 12 hours. For half the time the rate of vehicles was uniformly decreased from 5 to 0 vehicles per second, and for the other half the rate was uniformly increased from 0 to 5 vehicles per second. This signified peak times in the mornings and evenings, with less congestion during the middle of the day. At each iteration the emergence time was shifted by an amount taken from a Gaussian distribution with a deviation of 10 minutes. Further, an additional 10% vehicles were added to introduce some non-recurrent nature in the traffic system. Learning was carried out for a total of 10 iterations with a learning rate of 0.4. The value of σ_{min} was fixed to 7 miles/hour.

For testing, the same recurrent vehicles were used with some additional vehicles bringing in a non-recurrent aspect. Now the time to reach the goal was specified. The vehicles attempted to find the best route as well as the start time. The simulations were repeated for a number of vehicles over a number of scenarios. The experiments were further repeated for a number of values of α.

Each vehicle had its own start time, ideal finish time, time the vehicle actually finished its journey and travel time which were specific to its source, destination and ideal finishing time. The deviation of the vehicle arrival time at its goal was used as a metric of study. The ideal value is zero which would mean that a vehicle arrived at its destination at the time it was supposed to. A positive value specifies that the vehicle was late and the magnitude specifies the amount of time by which the vehicle was late.

For every value of α, a histogram showing the percentage of vehicles for the duration is given by Figure 3. The histogram is produced in pockets of 50 seconds. The average travel time for any vehicle was in the order of 15 minutes. The positive region of the graph shows late arrival which is undesirable. As expected, the higher values of α shift the histogram towards the negative region signifying that the vehicle reaches its destination much earlier than the expected time. This though causes a reduction in the percentage of vehicles arriving late.

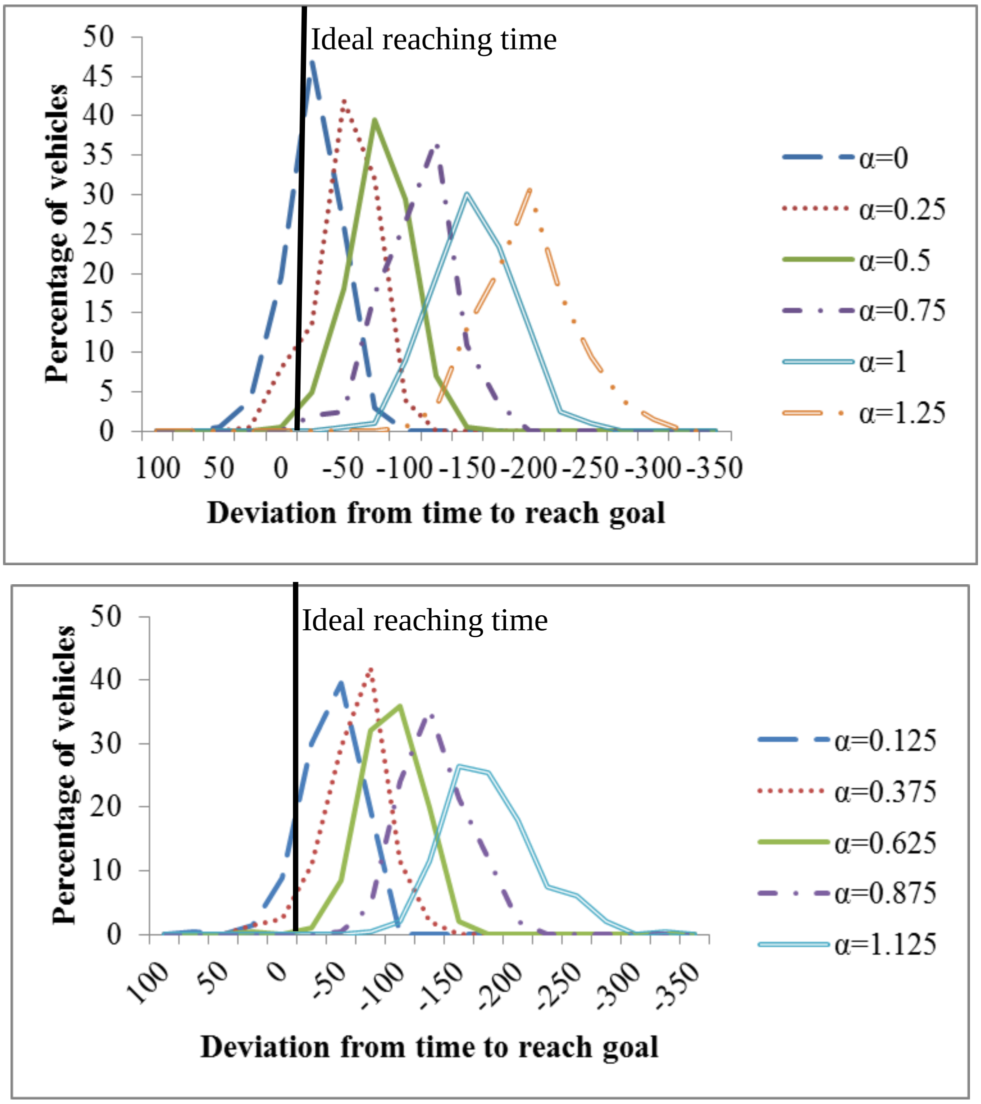

Figure 3: Histogram for deviation from time to reach the goal for the transportation system

Figure 4 specifically shows the percentage of vehicles arriving late for different values of α. The percentage is high for very low values of α, while it dies off very quickly as α increases. The mean deviation from time to reach the goal for different values of α is shown in Figure 5. Higher values of *α* make an average vehicle arrival time very early.

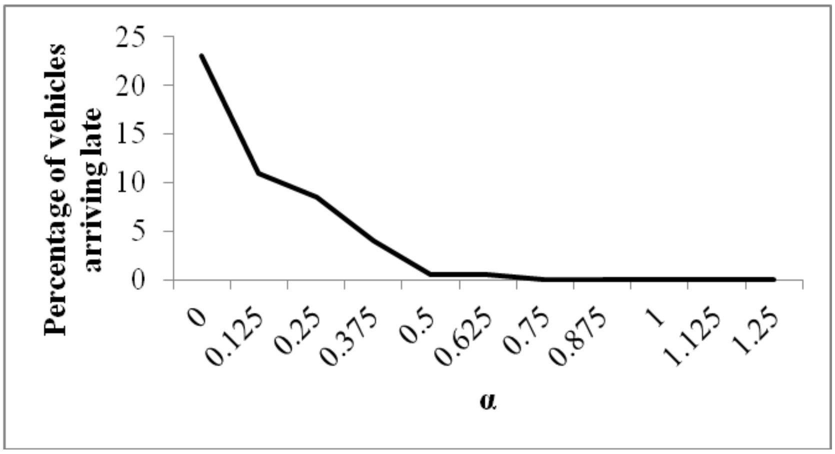

Figure 4: Percentage of vehicles arriving late

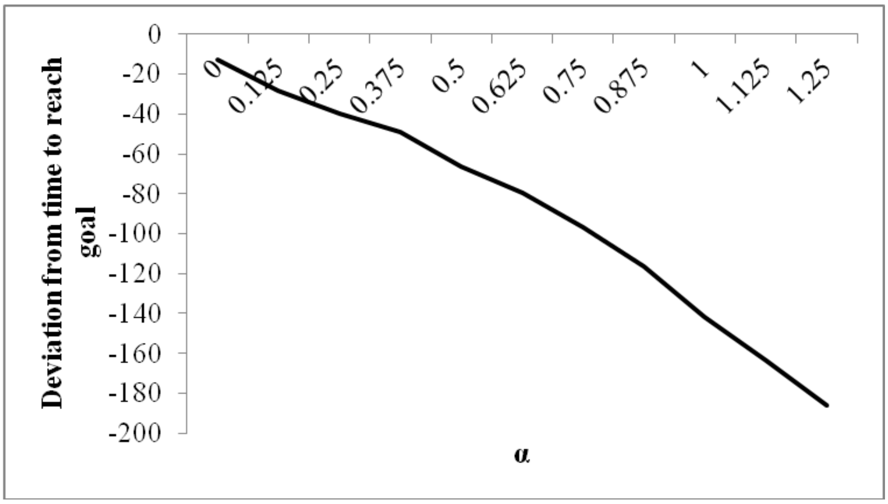

Figure 5: Deviation from time to reach the goal

Based on Figure 4 and Figure 5 it can be seen that a higher value of α minimizes the chances of arriving late, however at the same time the vehicle may arrive too early and may have to wait at the destination. A lower value of *α* is preferable when it is not very important to reach the destination either on time. Mostly one can take a chance of 10%-15%. Taking such a risk gives a big boon of departure time and the expected time to wait at the destination in case of early arrival is very small. However on occasions the 10%-15% risk might be very high. One may not always be ready to miss an important flight. It is perhaps best to take moderate values of α using which the risk may drop to 5%. The need to be assured of reaching the destination on time is (usually) a rare requirement, in which case one must be willing to leave very early. There is a large region in which the risk stays of the order of 1%-5%, implying that the typical result will be a very high waiting time at the destination.

This is in fact an experimental verification of a naturally observed phenomenon of how humans decide their start time and route under known traffic conditions. Arriving just before a regular meeting is common. Similarly, waiting for a long time for a flight is common. What we have done here however is to automate such procedures, at all times keeping the individual requirements of travellers to the fore. The proposed system is certainly better than using a system which gives some estimate of travel time with the user required to further guess the additional time as a precaution to traffic uncertainties. The guess is based on principles which are unreliable.

6. Conclusions

For everyday travel an individual is faced with the decisions as to their start time and the route of travel, which as a problem is not trivial. The algorithm presented in this paper attempts to solve the problem by exploiting the recurrent nature of traffic. The learnt information was used in computing the latest start time, at the same time ensuring that the probability of reaching the destination on time was as specified by the user. It was observed that one is likely not to need to wait for very long on average, provided he/she is ready to take a little risk in arriving late.

In the future, research may be carried out to better model the trade-off between the probability of reaching the destination and the lateness of departure, better accounting for non-recurrent traffic, enabling detecting and avoiding non-recurrent traffic, and an integration with transportation authorities. Computational limitations at present do not allow for learning and simulating a very large number of vehicles. Further the model needs to be validated using real traffic data, ultimately in a real-time, on line setting. The notion and factors of similar days, δ, learning rate and σ_{min} also need to be studied and accordingly set based on real traffic data for different traffic conditions. The currently applied decentralized approach requires a large number of connections which can be time-consuming. Hence colonies of centralized agents with decentralized inter-colony communication may be implemented in the future.

Acknowledgement

The authors wish to thank the Commonwealth Scholarship Commission in the United Kingdom and the British Council for their support of the first named author through the Commonwealth Scholarship and Fellowship Program (2010) - UK through award number INCS-2010-161.

References

1. Gordon, R.: ‘Chapter 3: Functional ITS Design Issues’, in ‘Intelligent Freeway Transportation Systems’(Springer, NY, 2009), pp. 17-40.
2. Kim, S., Lewis, M. E., and White III, C. C.: ‘Optimal vehicle routing with real-time traffic information’, IEEE Trans. Intell. Transport. Syst., 2005, 6, (2), pp. 178- 188.
3. Claes, R., Holvoet, T., and Weyns, D.: ‘A Decentralized Approach for Anticipatory Vehicle Routing Using Delegate Multiagent Systems’, IEEE Trans. Intell. Transport. Syst., 2011, 12, (2), pp. 64-373.
4. Weyns, D., Holvoet, T., and Helleboogh, A.: ‘Anticipatory Vehicle Routing using Delegate Multi-Agent Systems’, in Proc. IEEE Intelligent Transportation Systems Conference, Seattle, WA, 2007, pp.87-93.
5. Maniccam, S., ‘Adaptive decentralized congestion avoidance in two-dimensional traffic’, Physica A: Stat. Mech. Appl., 2006, 363(2), pp. 512-526.
6. Ando, Y., Fukazawa, Y., Masutani, O., Iwasaki, H., and Honiden, S.: ‘Performance of pheromone model for predicting traffic congestion’, in Proc. Fifth Int. Jt. Conf. on Autonomous Agents and Multiagent Systems, New York, 2006, pp. 73 – 80.
7. Narzt, W., Pomberger, G., Wilflingseder, U., Seimel, O., Kolb, D., Wieghardt, J., Hortner, H., and Haring, R.: ‘Self-Organization in Trafric Networks by Digital Pheromones’, in Proc. IEEE Intelligent Transportation Systems Conference, 2007, Seattle, WA, pp.490-495.
8. van Hinsbergen, C.P.I.J., Hegyi, A., van Lint, J.W.C., and van Zuylen, H.J.: ‘Bayesian neural networks for the prediction of stochastic travel times in urban networks’, IET Intell. Transport Syst., 2011, 5, (4), pp. 259-265.
9. Dia, H.: ‘An object-oriented neural network approach to short-term traffic forecasting’, Eur. J. Oper. Res., 2001, 131, (2), pp. 253-261.
10. S. Innamaa: ‘Short-term prediction of travel time using neural networks on an interurban highway’, Transportation, 2005, 32, pp. 649–669.
11. Kirby, H. R., Watson, S. M., and Dougherty, M. S.: ‘Should we use neural networks or statistical models for short-term motorway traffic forecasting?’, Int. J. Forecast., 1997, 13(1), pp. 43-50.

12. van Lint, J.: ‘Incremental and online learning through extended kalman filtering with constraint weights for freeway travel time prediction’, in Proc. Intelligent Transportation Systems Conference, Toronto, 2006, pp.1041-1046.
13. Miller-Hooks, E., and Mahmassani, H.: ‘Path comparisons for a priori and time-adaptive decisions in stochastic, time-varying networks’, Eur. J. Oper. Res., 2003, 146, (1), pp. 67-82.
14. Opasanon, S., and Miller-Hooks, E.: ‘Multicriteria adaptive paths in stochastic, time-varying networks’, Eur. J. Oper. Res., 2006, 173, (1), pp. 72-91.
15. Song, Q., and Wang, X.: ‘Efficient Routing on Large Road Networks Using Hierarchical Communities’, IEEE Trans. Intell. Transport. Syst., 2011, 12(2), pp. 132-140.
16. Tatomir, B., and Rothkrantz, L., ‘Hierarchical Routing in Traffic Using Swarm-Intelligence’, in Proc. 2006 Intelligent Transportation Systems Conference, Toronto, 2006, pp.230-235.
17. Li, Q., Zeng, Z., and Yang, B.: ‘Hierarchical model of road network for route planning in vehicle navigation systems’, IEEE Intell. Transport. Syst. Magz., 2009, 1, (2), pp. 20-24.
18. Bishop, R.: ‘Intelligent vehicle applications worldwide’, IEEE Intell. Syst. Appl., 2000, 15, (1), pp. 78-81.
19. Reichardt, D., Miglietta, M., Moretti, L., Morsink, P., and Schulz, W.: ‘CarTALK 2000: safe and comfortable driving based upon inter-vehicle-communication’, in Proc. 2002 IEEE Intelligent Vehicle Symposium vol. 2, 2002, pp. 545- 550.
20. Kala, R., ‘Motion Planning for Multiple Autonomous Vehicles’, PhD Thesis, University of Reading, UK, 2013.
21. Hart, P. E., Nilsson, N. J., Raphael, B., ‘A Formal Basis for the Heuristic Determination of Minimum Cost Paths’, IEEE Trans. Syst. Sci. Cybern., 1968, 4(2), pp. 100–107.
22. Openstreetmap, available at openstreetmap.org, accessed August, 2012.
23. M. Treiber, A. Hennecke, D. Helbing, Congested traffic states in empirical observations and microscopic simulations, Physical Review E 62(2): 1805–1824.