MAUTC Report

Topic:

Time Dependent Estimation of Travel Time in Real Time in Urban Areas

by

Antoine Hobeika
The objective of this project is to quantify and evaluate a wide range of methods that would estimate the travel times in urban areas under real time conditions taking into consideration that the traveler crosses the links of his and her route at different points in time, thus initiating the time dependency situation.

**Key Words**
- Travel times
- Congestion
- Network configuration

**Distribution Statement**
No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161
1. Introduction and Literature Review

The rapid growth in the application of Advanced Travel Information Systems (ATMS) requires the estimation of travel time on Arterial streets in real time. The travel time real-time estimation can help road users to make smart decision on route choice and avoid traffic jam in order to reduce the total travel time. Also, the travel time is an important parameter for evaluating the operation performance of the traffic network. A reliable travel time estimation can provide the needed information to the operator in order to optimize the traffic system.

Statistical methods, AVI technology, and traffic flow theory have been successfully used to estimate the travel time on freeways. However, few approaches exist to estimate real time travel times on an arterial street network with the complexity caused by signalized intersection.

Statistical methods have traditionally built a linear or a non-linear model, based on the historic traffic data, to predict the travel time on freeway or arterial streets. H. M. Zhang(1998)\(^1\) used the historic critical v/c ratio, occupancy from the loop detector to build up a non-linear model named “the journey Speed Model” to predict the travel time on arterial streets. The journey speed is represented as the weighted sum of the historic speed and the current speed from the detector.

Neural networks are models designed to imitate the human brain through the use of mathematical models. Similar to statistical methods, Neural Networks are built using previous existing data. However, neural networks can perform better than statistical methods in mapping the relationships between the travel times and the input data. In the past several years, neural networks have been successfully applied to predict short-term traffic flow and travel time. Abdulhai, Porwal and Recker(1999)\(^2\) used an advanced Time Delay Neural Network (TDNN) model combined with Genetic Algorithm (GA) to predict the traffic flow and density which can be used to estimate the travel time. Ishak, S., and C. Aleksandru (2003)\(^3\) used multiple topologies of dynamic neural network to optimize the short-term travel time estimation and prediction. They also tested and compared four different neural network architectures under different settings and traffic conditions.

The Webster formula\(^4\) introduces how to calculate the average signal delay near the stop line of an intersection. Later, Highway Capacity Manual 2000\(^5\) introduces the intersection control delay caused by the signalized intersection. The dynamic flow method and queuing method are used to compute the average delay of the vehicles caused by the red line. The intersection control delay includes uniform control delay, incremental delay (random arrivals and oversaturation queues) and initial queue delay. . Tsekeris and Skabardonis (2004)\(^6\) compared the performance of the method in HCM2000 with the spot speed model(SSM) and BPR methods on arterial networks. In their study, method in HCM2000 demonstrated the most promising modeling approaches.
Fadhely, Kenneth, and Donald (2002) introduced and compared queue methods from SIDRA, HCM2000, TRANSYT-7F, SOAP, NCHRP 279 Guidelines, SIGNAL 97, NETSIM and Oppenlander’s Method. In their study, the SIDRA and HCM2000 method have better accuracy than other methods since most of other methods will forgive the residual queue built up by the previous cycle length and some of them only report average value of the queue. Their research shows that it is important to consider the queue built up by the pervious time intervals.

The HCM2000 approach can estimate the average network performance in a relatively long time interval. However, it is not suitable for dynamic traffic applications in a short term time interval especially under the congested traffic situation.

Ashish Sen (1996) addressed the shortcomings by introducing the use of loop detectors to estimate the travel time. The loop detector is usually installed fairly close to the stop line of the intersection in order to help control traffic signal. As a result, the queue will easily build over the loop detector. Hence, the reading of the loop detector to obtain the number of vehicles and the speed will not be reliable and cannot reflect the queue length and how many vehicles are arriving at the intersection. If the detector is placed far from the stop line where the queue never extends over the detector, the travel time estimation requires the information about intersection capacity and the phases of the cycle length. Sungho Oh, Bin Ran and Keechoo Choi (2003) performed a study to find the best location of a detector to estimate the travel time on a relatively long urban link. They have utilized CORSIM in their study and they found that the optimal detector location was mostly related to link length and green time. Choi and Chung(2001) fused the travel time detected by GPS and loop detector to enhance the accuracy of the estimation procedure. The voting technique, fuzzy regression and Bayesian pooling method are utilized in the introduced fusion procedure.

A new category of arterial travel time estimation is based on the idea of tracking the same vehicle traveling along the arterial streets by GPS or by the vehicle identification technology. Similarly, vehicle platoon is identified and tracked in order to estimate the travel time by Lucas and Verma(2004). ITS Orange Book (2004) introduced the application of several technologies to estimate and predict travel times in networks. The study from University of Southampton Transportation Research (United Kingdom) shows that getting traffic information from a higher proportion of vehicles with GPS tracking system may not lead to an increase in estimated accuracy of travel times. Information from 26 percent of vehicles has the perfect performance. Their research provides a good guideline to select observed proportion of vehicles when using probe to estimate the travel time. Vehicle Relayed Dynamic Information (VERDI) uses the GSM network to communicate the probe vehicles with the control centers and determines their positions via GPS. It successfully combines the GPS and GSM technology to get the real time traffic information. Even though these approaches have very high accurate travel time estimation, it is too expensive to apply these advanced systems. In addition, these approaches are difficult to be used in the real-time travel time estimation due to the sample size issue.
Henry and Wenteng (2006) explored a time-dependent travel time estimation model for signalized arterial network. A dummy vehicle was traced to calculate its delays at every intersection at the time arrival in order to estimate the time-dependent travel time. The time-dependent travel time in the study is the recursively aggregation of the travel time on each link along the corridor.

Recently, many urban transportation departments are planning to place the detector far away from the stop-line in order to monitor traffic and provide travel time information. This deployment provides a good opportunity to utilize the detected traffic data to estimate the real-time travel time on the arterial network in one or two minutes update. HCM2000 has been the recommended method to estimate the intersection control delay. However, it has a shortcoming in estimating the initial delay in a relatively short-time interval update as discussed earlier. This thesis plans to modify the algorithms in HCM2000 to develop new algorithms for travel time estimation on arterial streets in real time.

2 Developed Algorithms

The methodology is separated into two sections. The first section introduces the algorithms to estimate travel time on an isolated arterial street. The second section introduces the methodology to estimate the travel time depending on different traffic situations considering the traffic situation on upstream and downstream streets.

2.1 Travel Time Estimation on an Isolated Arterial Street

The travel time on an isolated arterial street includes link travel time and intersection control delay. The link travel time can be computed based on the detected speed of the detector and the length of the street. The estimation of intersection control delay is more complicated and a dynamic flow algorithm is developed in the study in order to estimate the control delay. In the developed algorithm, the computation updated time interval is the signal cycle length of the intersection. The algorithms will estimate the average link travel time and intersection control delay of the observed vehicle group. The observed vehicle group is identified as the group of vehicles detected by the detector during the time interval when there is no blackout. If a blackout condition exists, the observed vehicle group is the group of vehicles detected by the detector on the upstream link during the time interval.

An intersection was observed in the study and the control delay analysis which shows that the control delay of the observed vehicle group has a strong relationship with the arrival time of the first vehicle of observed vehicle group as shown in Figure 1 and Figure 2. In addition, the percentage of vehicle in the observed group arrived the intersection during the red time has a significant effect on the intersection control delay which is similar with PF factor used in the HCM2000. Therefore, the algorithms will consider these two variables as well as volume to estimate the intersection control delay.
It is necessary to estimate the characteristics of the observed group to compute the stopped delay of the queued vehicles in the observed group while attempting to dissipate through the intersection. The stopped delay will vary with the size of the initial queue, the percentage of arrival in red time, the time the first vehicle in the observed group arrives at the red time and the volume of the observed group. In the algorithms in HCM2000, when the incoming volume is greater than the intersection capacity, uniform delay and over-saturation delay are computed independently to estimate the intersection control delay of the observed group. However, this separation is not needed to estimate the average stopped delay of the observed group behind the intersection, particularly if the ‘queue vs time’ curve is utilized. Therefore, in this algorithm, the computation of over-saturation and uniform delay is not separated.

The following relationships and assumptions are adopted:
a) The control delay is the average control delay of all the vehicles in the observed group.

b) The first vehicle in the first observed group will arrive at the intersection at the start of the red time phase.

c) The observed group will use a full cycle length to pass the intersection irrespective of what time the first vehicle in the group arrives at the intersection. This is consistent with the approach that we are trying to determine the travel time it takes for this observed group to traverse the link. This assumption is used in calculating the uniform delay for this observed group.

d) If the number of vehicles in the last observed group (for example Group 1 in Figure 3) is greater than the capacity of the intersection, some vehicles are queued at the intersection till the next green time takes place. Based on the aforementioned items c and d, the first vehicle of next observed group (for example Group 2 in Figure 3) will also arrive at the end of the queue at the start of red time phase.

Before the intersection control delay computation, the initial delay of the observed vehicle group is needed to be determined first. The initial delay of the first vehicle in the observed group (d3) is an important value in the developed algorithm. Different from the one computed in HCM2000, the initial delay of the first vehicle in the observed group in our case, as shown in Figure 4, is the time it takes the first vehicle in the observed group to travel from the time it arrives at the end of the initial queue to the time it arrives at the intersection. Our method computes the clearing time for the initial queue and not the average delay per vehicle in the queue.
Different from HCM2000, the adopted approach uses the shockwave method to estimate the initial delay of the first vehicle in the observed vehicle group and the initial queue.

If there is no blackout taking place on the loop detector which means that the traffic data observed by the loop detector is reliable, the shockwave method is used to estimate the initial queue length and then estimate the initial delay. The shockwave method utilized here is the same one used in the freeway case.

There are three cases for estimating the initial delay: a building queue case, a dissipation queue case and no change of queue case.

**Step 1:**

\[
W_u = \frac{V - C}{k_{td} - k_q}, \quad \text{for the building queue case where } V > C \\
W_d = \frac{C - V}{k_q - k_{td}}, \quad \text{for the dissipation case where } V < C; \\
W = 0, \quad \text{for no change in queue case where } V = C;
\]

Where

- \( V \): Incoming Volume;
- \( C \): Intersection Capacity;
- \( k_{td} \): is the vehicular density obtained from the detector;
- \( k_q \): is the jam density.

The queuing rate \( QR \) (veh/h) is:

\[
QR = \frac{dn}{dt} = (V - C - W_u \times k_{td});
\]

**Step 2:**

The number of vehicles in queue which is built during the observed cycle length \( t \) will be:
\( Q_m = QR \times CL \)

Where

\( CL \): is cycle length;

The process is repeated for every interval and the total number of vehicles in queue is estimated using the following expression:

\[ Q_t = \max(0, \sum_{m=1}^{i} Q_m) \]

\( Q_t \) is the initial queue for next cycle length which is also the queue at the end of cycle \( t \).

Thus, if we want to estimate the initial delay for cycle length \( t \), we should consider the queue at the end of last cycle length which is \( Q_{t-1} \).

If \( \frac{Q_{t-1}}{D_s} < g \), the initial delay is \( \frac{Q_{t-1}}{D_s} + \text{red}_{-\text{time}} \) (\( D_s \) is the departure saturation flow in green time veh/h/ln).

Else

If \( \frac{Q_{t-1}}{D_s} < 2g \), the initial delay is \( \frac{Q_{t-1}}{D_s} + 2 \times \text{red}_{-\text{time}} \); (because it should wait for another green time to dissipate the initial queue vehicle for the incoming group of vehicles in time \( t \))

Else

If \( \frac{Q_{t-1}}{D_s} < 3g \), the initial delay is \( \frac{Q_{t-1}}{D_s} + 3 \times \text{red}_{-\text{time}} \);

Thus, we came out with the following equation to estimate the initial delay.

When \( (n-1) \times g \leq \frac{Q_{t-1}}{D_s} \leq n \times g \),

Therefore, \( d3 = \frac{Q_{t-1}}{D_s} + (n) \times \text{red}_{-\text{time}} \)
After computing the initial delay for the previous time intervals, the ‘queue vs time’
curve can be determined to compute the intersection control delay. We have three cases
that describe variations in queue size and volume in determining the stopped delay which
is regarded as intersection control delay. They are:

Case 1- where there is no initial queue for the observed group.
Case 2 – where \( d_3 \) is smaller than a Cycle Length
Case 3- where \( d_3 \) is greater than a Cycle Length

**Case 1- There is no initial queue for the observed group.**

Figure 5 shows the development of the queue size in vehicles of the observed vehicle
group over time at a signalized intersection. The total stopped delay of the observed
group is made up of three components: 1) the stopped delay during the red time, 2) the
queue delay during the green time, and 3) the over-saturation delay which occurs in the
next cycle length.

![Figure 5 - Case 1- no initial queue for the observed group](image)

As shown in Figure 6, time \( t_0 \) to \( t_3 \) is observed cycle length. Time \( t_0 \) to \( t_2 \) is the red
phase and the time from \( t_2 \) to \( t_3 \) is green phase. The average stopped delay of the vehicles
detected within this cycle length represents the intersection control delay of this cycle
length. The first vehicle in the observed vehicle will arrive at the intersection at time \( t_1 \)
which can be detected by the detector. After time \( t_1 \), the queue will grow during the red
phase to include the incoming vehicles from the observed group. The arrival rate is
computed as the number of vehicle arrived during the red time divided by the remaining
red time which is \( t_2-t_1 \). At the end of the red time, the queue is built to \( k \). After time \( t_2 \),
the queue will decrease till the end of the green phase (\( t_3 \)) and now reaches the level of \( i \).
If volume is greater than the capacity, the value of \( i \) is greater than zero, otherwise, it is
equal to zero. The queue of the intersection will start to build up again during the red
phase. However, we need to separate the status of the vehicles in the observed group to
the status of what is happening at the intersection as a whole. In that regard, we separated
the queue development into a) the queue of the observed group and b) the queue at the
intersection. The queue of the observed group, which is our main interest as shown in Figure 6, remains constant during the red phase, and then dissipates again during the next green phase using a time (t5-t4).

The angle $a$ in Figure 6 represents the vehicle arrival rate from time $t_1$ to $t_2$. It can be computed as \[ \frac{k}{t_2-t_1} = \frac{P \times N}{t_2-t_1} = \frac{P \times N}{r-t_1}. \]

Where:

- $P$: the percentage of arrival vehicle in the red time;
- $N$: the total number of the vehicle in the observed vehicle group.
- $r$: red time of the intersection (secs);
- $g$: green time at the downstream intersection for the traffic movement under consideration (secs);

The arrival rate during green time is computed as $h_2 = \frac{g}{(1-p) \times N}$. The dissipating rate during the green time is assumed as $h_1$ (2 sec/veh) according to the field observation. Therefore, the queue dissipating speed ($h_3$) during the green time is equal to \[ \frac{1}{1/h_1 - 1/h_2}. \]

As a result, the value of $i$ in Figure 6 is computed as $i = k - g / h_3$. In additional, the queue of the observed group dissipating speed is equal to $h_1$. Since all the variables in Figure 6 are obtained, we can compute the total stopped delay of the observed vehicle group by computing the area under the curve shown in Figure 6.

Area $A$ is computed as follows:

\[ a = 0.5 \times (k) \times (r-t_1) \]

Areas $B$ and $C$ are computed as follows:

\[ b = 0.5 \times (i + k) \times g \]

\[ c = 0.5 \times (i \times h_1) \times i + i \times r \]

The average stopped delay for the observed group which is regarded as the intersection control delay for this case is \[ \text{delay} = \frac{a + b + c}{V \times CL}. \]

The part of reducing speed delay when the vehicle is approaching the intersection will be computed in the travel time on the link instead of the intersection control delay in our algorithm.
Case 2- there is an initial queue for the observed group and it is smaller than a cycle length

As shown in Figure 6, the queue will grow during the red phase to include the incoming vehicles from the observed group. At the end of the red time, the queue is now \( k + i' \). It will take a time of \( g_1 \) from the green phase to clear the initial queue. At this time the queue size is \( m \) because some additional vehicles have joined the queue from the observed group. In addition, all vehicles in the queue are from the observed group at this time. The queue will still decrease till the end of the green phase \((g_1 + g_2)\) and now reaches the level of \( i \). As described in case 1, the queue of the observed group, which is our main interest, remains constant during the red phase, and then dissipates again during the next green phase using a time \((t_6 - t_5)\).

The total stopped delay encountered by the observed group during time the initial queue dissipated \((d_3)\) is shown as area D in Figure 7. In addition, used variables such as \( h_1, h_2, h_3, \) and \( i \) are the same ones used in computing the case 1 of stopped delay. Therefore, the area D, which represents the stopped delay by initial queue, can now be calculated as follows:

\[
d = 0.5 \times (r - t_1) \times m + 0.5 \times (m + k) \times g_1
\]
\[ m = k + \frac{N \times (1 - P)}{g} \times g1 \]

Area A is computed as follows:
\[ a = 0.5 \times (i + m) \times g2 \]

Areas B and C are computed as follows:
\[ b = i \times r \]
\[ c = 0.5 \times (i \times h1) \times i = 0.5 \times h1 \times i^2 \]

The average stopped delay for the observed group for this case is
\[ delay = \frac{a + b + c + d}{V \times CL} \]

**Case 3- Initial queue clearance time (d3) is Greater Than the Cycle Length**

Since initial queue for the observed group is greater than the cycle length, the observed group will experience at least two red times and a green time before it arrives at the intersection. All vehicles in the observed group are assumed to be in the queue before they begin to depart the intersection. The queue of the observed vehicle group vs time is shown in Figure 8. As shown in Figure 8, the first vehicle in the observed vehicle will arrive at the intersection at time \( t_1 \) which is \( t_1 - t_0 \) after the beginning of the red time. After time \( t_1 \), the queue will grow during the red phase \( t \). At the end of the red time, the queue of the observed vehicle group is built to \( k \). After time \( t_2 \), the queue of the observed vehicle group will continue to increase till the end of the green phase (t3) and now reaches the level of N which is the number of vehicles in this observed vehicle group since no vehicle in the observed vehicle group can dissipate the intersection. After time \( t_3 \), the queue will remain the same till the first vehicle arrives at the intersection at time \( t_4 \). After \( t_4 \), the queue of the observed vehicle will decrease at the dissipating rate \( h1 \).
According to this assumption, the number of vehicles in the queue \((N)\) in Figure 8 is equal to the number of vehicle in the observed vehicle group. The dissipation rate of the observed group is \(h_1\) (2 sec/veh). From Figure 8, \(i = N - \frac{g^2}{h_1}\)

Area A is computed as follows:
\[
a = 0.5 \times (i + N) \times g^2
\]

Areas B and C are computed as follows:
\[
b = r \times i
\]
\[
c = 0.5 \times (i \times h_1) \times i = 0.5 \times h_1 \times i^2
\]

Area D is area under the curve from time 1 to time 4 which is computed as follows:
\[
d = 0.5 \times (t_2 - t_1) \times P \times N + 0.5 \times (P \times N + N) \times g + N \times (r + g_1)
\]

The average stopped delay for the observed group for this case is
\[
delay = \frac{a + b + c + d}{V \times CL}
\]

**Comparison of Results from Actual Delay vs Algorithm and HCM2000**

The real-field intersection control delay as well as the traffic volume are collected in order to compare the travel time estimation result with the developed methodology. In addition, algorithms in HCM2000 are utilized to compare the performance of the developed methodology. The following information was recorded in observed intersection: 1. arrival and departure time for every vehicle, 2. number of vehicles approached the intersection during each cycle length; 3 signal timing phases. The actual delay and the estimation results by developed methodology and methodology in HCM2000 are shown in Table 2 and Figure 8.

Table 2 Travel time in HCM2000 and Actual data Vs. the Algorithms travel time

<table>
<thead>
<tr>
<th>ACTUAL DELAY</th>
<th>HCM2000</th>
<th>ALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>58.45</td>
<td>61.69123</td>
</tr>
<tr>
<td>2</td>
<td>70.29</td>
<td>80.26934</td>
</tr>
<tr>
<td>3</td>
<td>57.64</td>
<td>69.40088</td>
</tr>
<tr>
<td>4</td>
<td>30.83</td>
<td>67.79689</td>
</tr>
<tr>
<td>5</td>
<td>44.00</td>
<td>48.90723</td>
</tr>
<tr>
<td>6</td>
<td>61.82</td>
<td>61.69123</td>
</tr>
<tr>
<td>7</td>
<td>45.80</td>
<td>59.24042</td>
</tr>
<tr>
<td>8</td>
<td>84.73</td>
<td>61.69123</td>
</tr>
<tr>
<td>9</td>
<td>33.43</td>
<td>53.31934</td>
</tr>
<tr>
<td>10</td>
<td>39.33</td>
<td>51.71618</td>
</tr>
<tr>
<td>11</td>
<td>77.82</td>
<td>80.26934</td>
</tr>
</tbody>
</table>
The Mean Absolute Error (MAE) is used to compare the fitness of two algorithm results with actual intersection control delay.

The Mean Absolute Error is calculated as: 

$$ MAE = \frac{1}{n} \sum_{i=1}^{n} |Observed - Estimated| $$

The Mean Square Error is calculated as: 

$$ MSE = \frac{1}{n} \sum_{i=1}^{n} (Observed - Estimated)^2 $$

Figure 8- Travel Times Comparison with Actual Delay, HCM2000 and Algorithm
The average MAE is around 10.85 seconds for the developed algorithm but 14.28 seconds for HCM2000. The MSE are 169.72 and 323.8 for the developed algorithm and HCM2000 respectively.

In addition, we use the statistical regression test to analyze the relationship with actual control delay with HCM2000 and developed algorithm. The NOVA tables are shown in below:

### Table 3 NOVA table for Actual Delay vs HCM2000

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>1</td>
<td>472.2</td>
<td>472.2</td>
<td>1.82</td>
<td>0.182</td>
</tr>
<tr>
<td>Residual Error</td>
<td>26</td>
<td>6733.4</td>
<td>259.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>27</td>
<td>7205.7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 4 NOVA table for Actual Delay vs Developed Algorithm

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>1</td>
<td>4267.4</td>
<td>4267.4</td>
<td>37.76</td>
<td>0</td>
</tr>
<tr>
<td>Residual Error</td>
<td>26</td>
<td>2938.3</td>
<td>113</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>27</td>
<td>7205.7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From the Table 3, the P-value is 0.182 which is greater than critical value of 0.05. Therefore, the result using HCM2000 is not statistically significant relative to the actual delay. However, the statistical test result shown in Table 4 is acceptable. It means the result using developed algorithm is statistically significant relative to the actual delay.

In general, the results show that the algorithms are robust and provide good accurate results when compared with HCM2000.

### 2.2 Approaches to estimate the travel time on Arterial Street considering downstream traffic conditions

The main structure of the overall approach to estimate the travel time on Arterial Street has four built-in algorithms, as shown in Figure 9. They are applied to cover the following traffic situations:

**Algorithm 1:** There is no bottleneck at the intersection.

**Algorithm 2:** There is a bottleneck at the intersection which is caused by a traffic jam in the downstream link.

**Algorithm 3:** There is traffic jam on link i and there is a blackout\(^1\) on the detector in the observed time interval.

\(^1\) **Blackout**: Blackout condition exists if a car stays over the detector for an extended period of time. This condition produces a high value of average dwelling time at the
**Algorithm 4**: Algorithm for a link \( i \) without an upstream link. Link \( i \) is the end link.

To start these algorithms we check whether there is a blackout on link \( i \). If yes, we use algorithm 3. If no, we can use algorithm 1 if the traffic situation has no or else algorithm 2 is used to cover the bottleneck traffic situation caused by link \( i+1 \). In addition, these algorithms consider the downstream conditions of the signalized intersection. Because the number of lanes on the downstream link as well as the vehicle queue of the downstream link will affect the travel time on the link under consideration. A drop in number of lanes in the downstream link and when the queue of the downstream link reach the intersection, it will reduce the dissipation rate at link \( i \) and consequently the travel time to traverse it.

The main functions of four algorithms are utilized to adjust the intersection capacity and the detected approaching volume to the intersection. In each algorithm, the travel time on an arterial link for a particular time interval is made up of two components. The first component is the delay encountered by the travelers at the signalized intersection which is referred to as intersection control delay, and the other component is the travel time on the regular stretch of the arterial link before arriving to the intersection. The control delay is very much dependent on the intersection capacity and its dissipation flow rate.

detector which is approximately 0.7sec/veh (obtained from CORSIM simulation). To increase the accuracy of the algorithm, a blackout situation is declared when the average dwelling time is over 0.6sec/veh.
2.2.1 Condition A—Algorithm 1:
Condition A applies when there is a normal traffic situation. That is, when there is no blackout on the loop detector and there is no bottleneck at the intersection. The flow chart of the algorithm is shown in Figure 10.
Algorithm 1 Flow Chart

$D_{si} \times N_i$ is the maximum departure volume during the green time on link $i$, and $S_{i+1} \times N_{i+1}$ is the maximum acceptable volume by the downstream link $i+1$. A lane drop or a lane closure on downstream link $i+1$ will cause the total maximum departure volume on link $i$ to be greater than the maximum total acceptable volume on link $i+1$. In this case, we should reset the maximum departure flow rate on link $i$ to be

$$D_{si} = \frac{S_{i+1} \times N_{i+1}}{N_i} \text{ (veh/h/ln)}.$$ 

The new capacity of link $i$ is

$$C = \frac{S_{i+1} \times N_{i+1} \times g}{N_i \times CL}.$$ 

Algorithm 1 represents two traffic conditions with two cases. If $Vp/C$ is greater than 1, the algorithm uses queue building method to determine the initial queue length and compute the intersection capacity introduced above, otherwise use queue dissipation method. In addition, the detected speed is used to compute the link travel time in this case. The total travel time includes the link travel time and the intersection control delay.

### 2.2.2 Condition B—Algorithm 2

In this algorithm, there may be a bottleneck at the intersection which is caused by a traffic jam at the downstream link $i+1$. If the bottleneck exists, the volume detected by the detector at link $i+1$ minus the volume of turning vehicles represents the maximum departure volume of link $i$.

In this condition, the volume detected by the loop detector on link $i+1$ is close to the actual capacity of this link particularly if the queue extends to within 100 feet of the end of this link. Figure 11 displays the flow chart of this algorithm.
Algorithm 2:

![Flow chart of Algorithm 2](image)

Is \( C \cdot L_{i+1} - Q_{i+1} > 100 \) ft?

- No
- Yes

\[
C = \frac{(V_{i+1} - V_t) \times (N_{i+1})}{N_i}
\]

Is \( V_p/C < 1 \)?

- Yes
- No

Where

- \( V_{i+1} \): the volume detected at link \( i+1 \);
- \( V_p \): traffic volume detected at link \( i \) during the previous time interval;
- \( V_t \): The turning on volume on this link minus the turning out volume on this link of this intersection
- \( S_{i+1} \): Acceptable flow on link \( i+1 \)
- \( L_{i+1} \): Length of Link \( i+1 \) in (ft).
- \( QL_{i+1} \): Length of vehicle queue in (ft).

The most important mission of algorithm 2 is to reset the capacity of link \( i \) based on the conditions of link \( i+1 \). Once the capacity of link \( i \) is reset, the estimation of travel time on link \( i \) is carried in the same way as in algorithm 1. The actual capacity of link \( i \) in this condition is \( C = V_{i+1} - V_t \). Where \( V_{i+1} \) is the volume detected by loop detector of link \( i+1 \), and \( V_t \) is the turning movement flow.

The delay and queue estimations are the same as algorithm 1 except for the new estimation of links \( i \) capacity.

2.2.3 Condition C—Algorithm 3

Condition C takes place when there is blackouts at the loop detector of link \( i \). In this condition, the data detected by the loop detector can not show the right incoming volume. Thus, we use the volume detected by its upstream link (link \( i-1 \)) compared to the capacity of the upstream intersection \( C_{i-1} \) as the incoming volume and the associated observed group of vehicles.

Let us assume that the queue length on link \( i \) do not reach the end of the link, and it is greater than 100 feet away from it. Let the volume detected by the upstream link is \( V_{i-1} \), the initial queue at link \( i-1 \) is \( Q_{i-1} \), and the capacity of the upstream intersection is \( C_{i-1} \). When \( V_{i-1} + Q_{i-1} \) is greater than \( C_{i-1} \) which means the volume is greater than the intersection capacity, then the maximum dissipated volume of link \( i-1 \) is \( C_{i-1} \) instead of \( V_{i-1} + Q_{i-1} \). Thus in this case, let \( V_{i-1} = C_{i-1} \), and this \( V_{i-1} \) represents the dissipating...
volume from the upstream link i-1 as shown in Figure 12. But the incoming volume to link i is \((V_{i-1}+ Q_{i-1})*(1+t\%)\), where \(t\%\) is the percent of accumulated volume from the turning movements at an intersection as explained earlier. But the incoming volume to link i is \((V_{i-1}+ Q_{i-1})*(1+t\%)\), where \(t\%\) is the percent of accumulated volume from the turning movements at an intersection as explained earlier.

Then, if \((V_{i-1}+ Q_{i-1})*(1+t\%)\)\(N_{i-1}<Dsi*N_i\), which means that the total saturation flow-restrained volume for link i is greater than the total incoming volume from link i-1, put

\[V_i = (V_{i-1}+ Q_{i-1})*(1+t\%)\frac{N_{i-1}}{N_i}\]

Where \(V_i\) represents the incoming volume for link i.

When the incoming volume to link i is greater than the saturation flow–restrained volume for link i volume, which is \((V_{i-1}+ Q_{i-1})*(1+t\%)\)\(N_{i-1}>Dsi*N_i\), then let the incoming volume and observed group be represented by the maximum acceptable flow rate which is \(Dsi(\text{veh.h/ln})\).

In case that the queue length in link i is within 100 feet of the end of link i, then the incoming volume to link i can not be greater than the discharge volume of link i, which is \(C_i\). So if \(N_i*C_i\) is greater than \(N_{i-1}*\max(C_{i-1}, V_{i-1}+ Q_{i-1})*(1+t\%)\), \(\max(C_{i-1}, V_{i-1}+ Q_{i-1})*(1+t\%)\)\(N_{i-1}/N_i\) prevails or else the incoming volume should be equal to \(C_i\).

Having determined the new observed group and the incoming volume for algorithm 3, then we can continue to use algorithm 1 and 2 to estimate the intersection delays and the travel time on link i.

Since we do not have the right density data on the loop detector, we can not use the shockwave method to estimate the queue length. The alternative method to estimate the change in queue length is to use \((V-C)\)\(CL\). If \(V\) is greater than \(C\), it is a queue building case; otherwise, it is a queue dissipation case.
Algorithm 3: Blackout on the loop detector in link i

Figure 12 - Flow chart of Algorithm 3

2.2.4 Condition D—Algorithm 4

Algorithm 4 is applied in the situation when the observed link i is the end link. That is, it does not have an upstream link. When there is a blackout on link i, we cannot utilize algorithm 3 to estimate the incoming volume since there is no upstream link. Thus, we assume the maximum travel time on this link where there is a blackout situation. The maximum travel time is computed as \[ \frac{L_i}{19} + \text{uniform delay} + \text{over-saturation delay}, \]
where \( L_i \) is the length of the link i (ft) and 19 is the average length of the car plus the jam headway in (ft). Figure 13 shows the flow chart of this algorithm.
2.2.5. Adjustment of the time interval updates algorithms

In some cases, the detector updated time is different from the intersection cycle length and can not be modified. Therefore, we need to adjust the volumes detected by the sensor according to the remaining time differences between the detection time and the cycle length time in order to achieve the traffic information during each cycle length.

This can be best illustrated first by using an example. Let us assume that we have an intersection with a cycle length of 100 secs, and the detected volume is updated at every 60 secs.

The volume at time period t between 101 and 200 secs is calculated as follows:

From diagram below the following inequalities hold:

\[ n \times 60 \leq (t-1)100 \leq (n+1)60, \text{ and} \]
\[ k \times 60 \leq (t)100 \leq (k+1)60, \]

where

- t is the observed time interval, and \( n \times 60 \) and \( (n+1)60 \) are the closest values to \((t-1)\times 100\),
- and \( k \times 60 \) and \( (k+1)60 \) are the closest values to \((t)100\).
- \( n \) and \( k \) are integer values.
To get the volume at time interval 101-200

Hence, the volume at time period \( t \) is:

If \( t = 1 \)

\[
V_t = V_{60} + V_{120} \times \frac{100 - 60}{60}
\]

otherwise

\[
V_t = \frac{[(n + 1) \times 60 - (t - 1) \times 100]}{60} \times V_{(n+1)\times60} + \frac{100 \times t - 60 \times k}{60} \times V_{(k+1)\times60} + (k - n - 1) \times V_{k\times60}
\]

To generalize the above procedure, the following equations are developed to address the various possibilities in Cycle Length (CL) for an update time \( t_{up} \) of 60 (secs) from the detector.

1. When update time \( \leq \) cycle length \( \leq 2*\)update times.

The equation is:

\[
n* t_{up} \leq (t-1) CL \leq (n+1)* t_{up},
\]

\[
k* t_{up} \leq (t) CL \leq (k+1) * t_{up},
\]

if \( t = 1 \)

\[
V_t = V_{k(t_{up})} + V_{(k+1)(t_{up})} \times \frac{CL - t_{up}}{t_{up}},
\]
Otherwise
\[ V_t = \frac{[(n+1) \times t_{up} - (t-1) \times CL]}{t_{up}} \times V_{(n+1)\times t_{up}} + \frac{CL \times t - t_{up} \times k}{t_{up}} \times V_{(k+1)\times t_{up}} + (k - n - 1) \times V_{k\times t_{up}} \]

2. When \(2 \times n \text{ update times} \leq \text{cycle length}\)

\[ n \times t_{up} \leq (t-1) \times CL \leq (n+1) \times t_{up}, \]

\[ k \times t_{up} \leq (t) \times CL \leq (k+1) \times t_{up}, \]

if \( t=1; \)

\[ V_t = \sum_{h=k}^{t} V_{h(t_{up})} + V_{(k+1)(t_{up})} \times \frac{CL - t_{up}}{t_{up}} \]

Otherwise;

If \(k-n-1=1\)

\[ V_t = \frac{[(n+1) \times t_{up} - (t-1) \times CL]}{t_{up}} \times V_{(n+1)\times t_{up}} + \frac{CL \times t - t_{up} \times k}{t_{up}} \times V_{(k+1)\times t_{up}} + (k - n - 1) \times V_{k\times t_{up}} \]

If \(k-n-1=2\)

\[ V_t = \frac{[(n+1) \times t_{up} - (t-1) \times CL]}{t_{up}} \times V_{(n+1)\times t_{up}} + \frac{CL \times t - t_{up} \times k}{t_{up}} \times V_{(k+1)\times t_{up}} + V_{k\times t_{up}} + V_{(k-1)\times t_{up}} \]

Hence, the final equation is:

\[ V_t = \frac{[(n+1) \times t_{up} - (t-1) \times CL]}{t_{up}} \times V_{(n+1)\times t_{up}} + \frac{CL \times t - t_{up} \times k}{t_{up}} \times V_{(k+1)\times t_{up}} + \sum_{h=0}^{k-n-1} V_{(k-h)\times t_{up}} \]

Travel Time Update at every 2 minutes (120 seconds):

If we chose to update the traffic data from the detector at 2 minutes interval instead of one minute interval, then the observed volume or vehicle group during the traffic light cycle length is determined by the following formulae:

If cycle length (CL) is assumed to be 100 seconds, the observed volume at the first cycle interval is:
For other cycle intervals, we adjust the detected volumes according to the remaining time differences between the detection time and the cycle length time. Following the previous example, the volume at time period $t$ is calculated as:

$$V_t = \frac{\left[ (n) \times 120 - (t - 1) \times 100 \right]}{120} \times V_{n \times 120} + \frac{100 - \left[ (n) \times 120 - (t - 1) \times 100 \right]}{120} \times V_{(n+1)\times 120}$$

This equation can be equally applied when cycle length < update time as;

3. cycle length < update time

If $t = 1$

$$V_{CL} = V_{up} \times \frac{CL}{t_{up}}$$

For all other time updates:

$$V_t = \frac{\left[ (n) \times t_{up} - (t - 1) \times CL \right]}{t_{up}} \times V_{n \times t_{up}} + \frac{100 - \left[ (n) \times t_{up} - (t - 1) \times CL \right]}{t_{up}} \times V_{(n+1)\times t_{up}}$$

3.2.6 Validation

Three different time interval updates by the loop detector are simulated in CORSIM to test the sensitivity of the results to various time updates based on the same intersection cycle length of 100 seconds and with the same incoming volumes. The three time updates are 60 sec, 100 sec and 120 sec respectively. The cycle length is 100 seconds. The travel times for different updated time intervals computations use the output of the simulated detector in CORSIM. The developed methodologies are used to compute the travel time. These travel times are used to compare with simulated travel time by CORSIM. The result is shown in Figure 14.
The average MAE is around 16 seconds. The biggest difference is in time interval 5 which has a 42 seconds difference with the CORSIM travel time. The incoming volume of the observed group in time interval 5 is about 50% over the capacity which will take an additional amount of time to clear the intersection after this interval. It will significantly increase the intersection control delay of the observed vehicle group. Since CORSIM only computes the average travel time at the end of this time interval, it doesn’t count the clearing time for the vehicles which can not depart the intersection at the end of this interval. Therefore, the results show that the algorithms are robust and provide good accurate results when compared with CORSIM.

Conclusion

Estimation of travel times on arterials has been a challenging task because vehicles traveling on arterials are not only subject to queuing delay but also subject to traffic signal delay. Few transportation professionals have conducted research at estimating travel times on arterial street networks, and even fewer of them have utilized the dynamic flow methods to estimate these travel times in a short updated time interval with the detector in the middle of the street.

This paper focuses on real-time estimating of arterial travel time by using the real-time detector data on arterial links. Algorithms have been developed and have been validated by the actual data as well as the developed algorithm in HCM2000. In addition, several time updates of the detector data: 100 seconds, 60 seconds, and 120 seconds were simulated in CORSIM to validate the time interval update algorithm.

In general, the statistics show that the developed algorithms are robust and provide good accurate results when compared with HCM2000.
Reference:

1. H.M. Zhang ‘A link journey speed model for arterial traffic’ Research Report, Civil and Environmental Engineering, University of California, Davis, December 1, 1998;


5. HCM2000, Highway Capacity Manual 2000 by Transportation Research Board;


