A PRATICAL IMPLEMENTATION OF TIME-DEPENDENT FASTEST PATH ALGORITHM: A CASE STUDY OF BANGKOK ROAD NETWEOK

Kriangsak Vanitchakornpong, Nakorn Indra-Payoong, Peerapol Sittivijan
Faculty of Logistics, Burapa University
eaksung@gmail.com, nakorn.ii@gmail.com, peerapoljt@gmail.com

Introduction

Shortest path algorithms e.g. Dijkstra and A* have been applied to business and industrial applications. Unit of measurement can be various in a shortest path algorithm. With distance, the algorithms try to find the path with shortest distance between a pair of origin and destination while in case of travel time, the algorithm try to find a path with minimum travel time or fastest path instead. In case of fastest path, the resulted fastest path would be reasonable when the variation of travel time on the network is small. However, when the variation of travel time is large the resulted fastest path would be unreasonable and inapplicable. For example, in large cities, travel time between peak and off-peak periods can be obviously different. An alternative is to apply time-dependent travel time e.g. a set of travel time during peak period and another one for off-peak period to the fastest path algorithms. However, the implementation of time-dependent fastest path is problematic when there are many periods to be considered. In this case, the computational complexity and required data to be processed can be enormous. In this paper, the pre-processing step and database architecture of time-dependent fastest path will be introduced. The pre-processing and database architecture were applied to Dijkstra algorithm to reduce computational time for time-dependent fastest path algorithm. The fastest paths for all pairs between origins and destinations in the network in each period were pre-calculated and stored in the database based on travel time in each period. These fastest paths will be retrieved and combined to find the fastest paths between an origin and destination at the specified starting time at origin node. The database is designed so that the pre-calculated fastest paths for all pairs between origins and destinations will be easily retrieved. In Section 2, related literature will be discussed. Section 3 describes the pre-processing and data architecture for time-dependent fastest path algorithm. Section 4 describes application of the pre-processing and data architecture proposed in Section 3 to the time-dependent fastest path with arrival time-window and to the time-dependent fastest path with multiple drops. Section 6 concludes all of the work and discusses potential future work.

Literature Review

Dijkstra (1959) proposed an exact algorithm to find a shortest path between an origin and destination. In Dijkstra's algorithm, the search begins by trying to find a node with shortest distance to the origin node and extends node by node in this fashion until reaching the destination node. Dijkstra's algorithm uses only the distance from the origin node in finding the shortest path. A* algorithm (Hart *et al*., 1968) improves Dijkstra's algorithm by using both the distance from the origin and the distance to the destination for a node to find a shortest path. Fu et al. (1998) proposed an algorithm to calculate expected fastest path where travel time on a link in the network is dynamic and stochastic. The calculation uses average and variation of travel to find an expected fastest path. However, this also adds complexity to the problem and it takes longer computational time. In this case, they proposed k-shortest path instead of shortest path to reduce computational time. Chen et al. (2005) proposed algorithms to find fastest paths under three different searching criteria, namely, expected value, dependent-choice and choice-constrained. Genetic algorithm was applied to find fastest path under these criteria Zhang et al. (2006) proposed allFP query method to find fastest paths with specified departure and arrival time interval. The allFP query method was developed based on A* algorithm. One of its advantages is it can keep track of previous calculated data, therefore, there is no need to run the algorithm every time when one needs to find the fastest paths. They also proposed the way to estimate the lower bound of the fastest path to reduce the computational time. Wu and Nie (2009) studied the algorithm to find fastest path in the network with stochastic travel time. They proposed alpha-discrete scheme which is able to find fastest paths in a large network without necessarily adjusting parameters for a specific network.

Pre-processing and Data Architecture for Time-dependent Fastest Path Algorithm
An example of time-dependent path can be illustrated in Figure 1. In this example, there are two routes i.e. R1 and R2 from the origin a to the destination d. For route R1, it passes through links: ab-bc-ce-ed and, for R2, links: ac-ce-ed. Common links between these two paths are ce and ed. As one can see that, for route R1, the traveler arrives at the beginning point of link ce at 8:20 and it takes 5 minutes to travel through link ce. In comparison, for route R2, the traveler arrives at the beginning of link ce at 8:10 and it takes 8 minutes to pass through link ce. In this case, to find a shortest path between an origin and a destination, one has to take into account the arrival time at the beginning of each link and the travel time of that link at that specific arrival time.

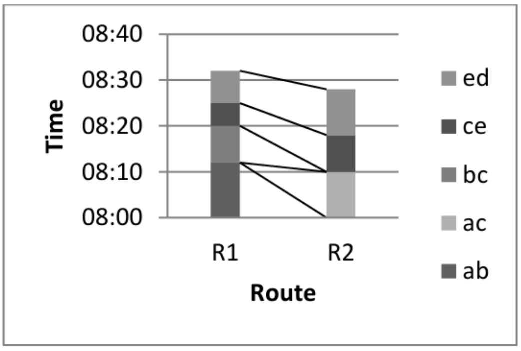


Figure 1: an example of time-dependent path

In order to consider the time-dependent travel time in the fastest path algorithm, the conventional mathematical programming to find shortest paths can be adjusted as follows.

$$min \sum_{i \in N} \sum_{j \in N} x_{ij} t_{ij}^{p} \quad (1)$$

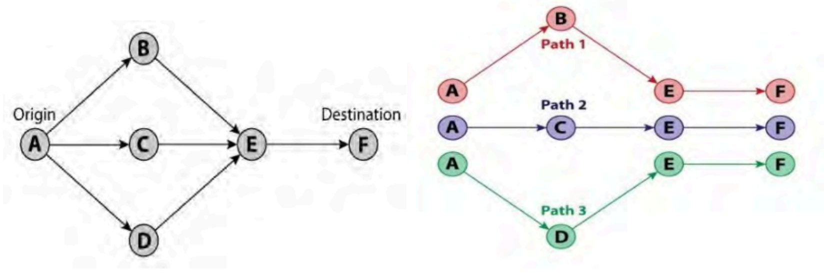
S.T.

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ij} = 0 \ ; \ \forall i \in N \setminus \{s, e\} \quad (2)$$

$$\sum_{j \in N} x_{sj} - \sum_{j \in N} x_{js} = 1 \quad (3)$$

$$\sum_{j \in N} x_{ej} - \sum_{j \in N} x_{je} = -1 \quad (4)$$

Where
N = set of all nodes in the network, x_{ij} = 1 if arc ij is a part of the shortest path; 0 otherwise, t_{ij}^{p} = travel time from node i to node j with staring time from node i equal to p, s = an origin node, e = a destination node
The objective function is to minimize the total travel time from an origin node to a destination node. Constraints (1) – (3) is the conventional constraints for network modelling. In this case, we assumed that $t_{ij}^{p_1}$ and $t_{ij}^{p_2}$ has the same value if p_1 and p_2 are in the same time interval, k. For example, if k is equal to 5 and the initial time is set to be 0, then the value of t_{ij}^{2} and t_{ij}^{4} would be the same, as well as, the same value between t_{ij}^{122} and t_{ij}^{124}. In addition, we also assumed that the link travel time satisfies the first-in first-out (FIFO) property, this implies that a vehicle departing later cannot arrive earlier than a vehicle departing earlier. The following example shows how to find a time-dependent fastest path for a small network.



a) b)

Figure 2: a) an example network b) 3 possible paths from A to F

Figure 2 shows a small network to illustrate how to find time-dependent fastest path. In the example, the origin is A and the destination is F. There are three possible paths from A to F e.g. through B, C and D. Figure 3 shows the travel time of each link during different time period. For example, the travel time on the link AC, t_{ij}^{p} when $0 \leq p < 5$ is equal to 4. However t_{ij}^{p} when 15$\leq p < 20$ is equal to 6. If one starts his/her journey from A to F at time 0 then the fastest path will be on the route A-B-E-F with the total travel time of 21 units.

Time horizon	AB	AC	AD	BE	CE	DE	EF
0	4	6	4	12	12	13	5
5	5	5	5	10	11	13	7
10	3	5	3	12	9	11	6
15	6	6	5	9	10	10	5
20	6	3	3	11	10	11	6
25	6	6	6	11	9	11	7

Figure 3: link travel time depending on the arrival time period of that link

However, if one starts the journey at time 5 and 10 then the fastest path will be A-C-E-F with the total travel time of 19 and A-D-E-F with the total travel time of 20, respectively. These fastest paths at different starting time from A can be illustrated in Figure 4

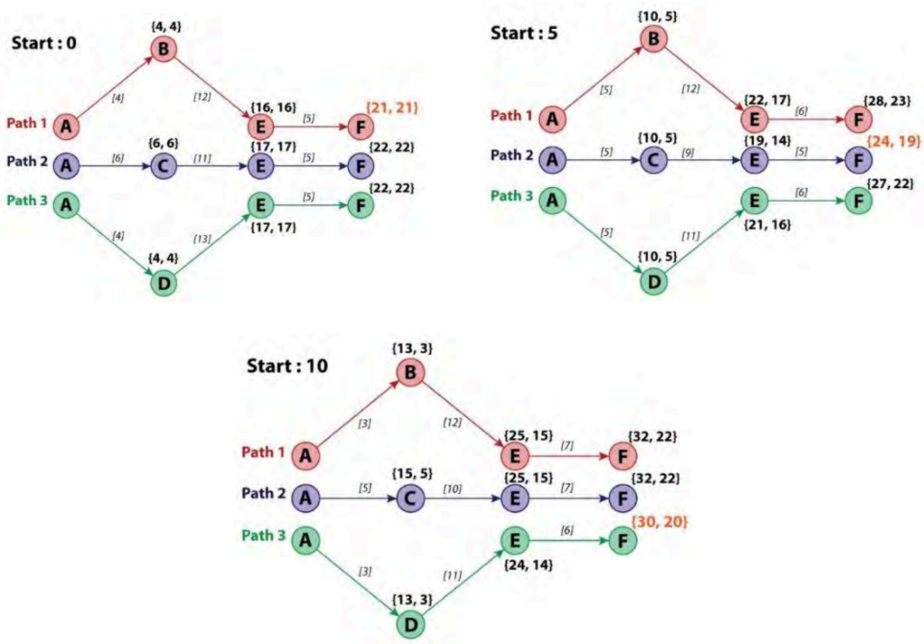


Figure 4: Fastest paths from A to F at different starting time.

As one can see, the computational time to find time-dependent fastest path can be exponentially intractable based on the curse of dimensionality. This could be problematic for real-time applications. Therefore, we proposed a pre-processing step to calculate the fastest paths for all origin-destination (O-D) pairs for all time periods in advance and store the pre-processed data in the database. To improve the computational time of time-dependent fastest path, the multi-threading calculation was applied to calculate time-dependent fastest path for each O-D in parallel. The pre-processed data of time-dependent fastest path can be fast and easily retrieved for the databased when it is required by the applications.

Empirical Study

To examine the efficiency of the time-dependent fastest path with the proposed pre-processing and data architecture, the experiments were conducted on the road network in Bangkok consisting of 3,500 × 3,500 nodes. In this paper, it was assumed that the interval of travel time is five minutes and the travel time, $t_{ij}^{p_1}$ and $t_{ij}^{p_2}$ are equal if p_1 and p_2 are in the same time interval. The time-dependent fastest paths for all origin e.g. 3,500 nodes to all destination 3,500 for all 288 time intervals e.g. 24 hours for every 5 minutes were calculated based in the proposed algorithm and store in the database to be retrieved and utilized. Figure 5 illustrates the travel time on the fastest path of a free-flow path from 6 AM to 7 AM. As expected, the travel time varies from 30 minutes to 45 minutes which is quite small.

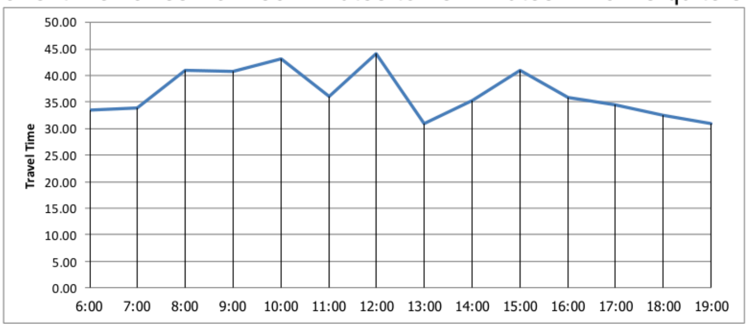


Figure 5: An example of travel time on the fastest path of a free-flow path from 6 AM to 7 PM

To better understand variation of travel time on time-dependent fastest path, the time-dependent fastest path of three O-D pairs, namely, O-D pair I, J and K on the Bangkok road network were examined. The mean and standard deviation values the travel time of the time-dependent fastest paths on these three O-D pairs were collected for every 30 minutes. For example, for O-D pair I, the travel time for the time-dependent fastest paths for every 5 minutes from 3.30 PM to 4 PM were calculated and the mean and standard deviation of travel time for O-D pair I from 3.30 PM to 4 PM were calculated.

In the experiment, the total costs comprising of vehicle operating cost (VOC) and value of time (VOT) between the shortest paths and time-dependent fastest paths for paths I, J and K. The travel time for both the shortest and time-dependent fastest paths were calculated based on the historical data on Wednesday for 50 days at 7 AM and 5 PM. The values of VOC and VOT used in the experiments were 5 Baht/KM and 300 Baht/hour, respectively. Table 1 shows the comparison of the total costs between the shortest and time-dependent fastest paths for all three O-D pairs.

Path	Peak	Shortest			Fastest				Savings	
	Time	Dist	Time	*Cost**	Dist	Time	Stdev	*Cost**	Baht	%
I	7:00	5	41.45	232	11.49	35.75	3.69	236	-4	-
	17:00	5	46.84	259	12.63	32.75	3.30	227	32	12
J	7:00	15	98.34	567	24.56	65.25	6.65	449	118	21
	17:00	15	101.45	582	25.36	84.00	14.02	547	35	6
K	7:00	33	151.61	923	55.14	89.75	16.78	724	199	22
	17:00	33	142.87	879	52.70	110.75	26.15	817	62	7

Table 1: Comparison of the total costs between shortest and time-dependent fastest paths

The mean and standard deviation values of travel time for the time-dependent fastest paths for all O-D pairs for all intervals were calculated and compared as shown in Table 2. In Table

2, the travel time for time-dependent fastest paths was calculated at 7 AM and 5 PM as well as ± 5, ± 10 and ± 15 minutes. It can be observed that, for these three O-D pairs, there is not much difference of the mean and standard deviation values of travel time in different time interval. Therefore, it might be possible that the range of interval can be increased to 15 minutes, for example, to reduce the total number of interval in a day and improve the efficiency of the computation.

Time-dependent Fastest Path with Arrival Time Window

In this section, the algorithm to find time-dependent fastest path will be applied to an additional condition i.e. a restricted arrival time window. This condition restricts the arrival time at the destination to be within a pre-specified time interval.

Path	Time	Mean travel time (min)			
		0	±5	±10	±15
I	7:00	40	44 (6.08)*	47 (5.86)	42 (8.79)
	17:00	31	41 (10.02)	43 (9.03)	44 (8.38)
J	7:00	63	61 (1.73)	61 (1.64)	62 (2.00)
	17:00	66	67 (2.31)	69 (3.32)	72 (6.40)
K	7:00	114	117 (8.50)	123 (18.10)	120 (15.55)
	17:00	135	128 (9.64)	125 (12.26)	124 (10.52)

Table 2: Comparison of travel time for time-dependent fastest paths in different time intervals

* Number in parenthesis is the values of standard deviation (min)

$$min \sum_{r \in R} x_r (e_r - s_r) \tag{5}$$

S.T.

$$\sum_{r \in R} x_r = 1 \tag{6}$$

$$l \leq e_r \leq u \tag{7}$$

Where

R = set of all possible routes from an origin node to a destination node, x_r = 1 if route r is selected and 0 otherwise, e_r = starting time for route r, s_r = arrival time for route r

For the small example in Figure 4, the time-dependent fastest paths with the starting time at 0, 5 and 10 can be shown in Figure 6. If the restriction on the arrival time at node F is between 20 and 25, then path 3 will be violated the constraint and the optimal path will be path 2 with the travel time of 19 units.

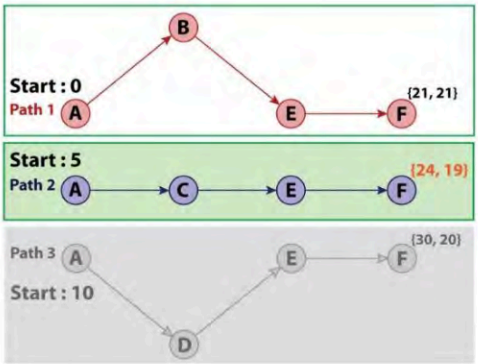


Figure 6: Time-dependent fastest path with different stating time

We also conducted an experiment to compare the travel time selected by the travelers and that calculated by time-dependent fastest path. A random sample of 30 people traveling between 3 O-D pairs, namely, L, M and N were asked about their selected paths on October 3, 2012. All these three O-D pairs were selected because they are all in the central business district and their distances are short. The mean values of travel time for the selected paths and time-dependent fastest path between these three O-D pairs from 10 AM to 7.30 PM were shown in Table 3.

O-D	Time window	Selected paths				Fastest Paths			
		Dist	Time	Depart	Arrival	Dist	Time	Depart	Arrival
L	10:00 - 10:30	23.69	49	9:25	10:14	24.23	42	9:35	10:17
	13:30 - 14:00	24.68	55	12:35	13:30	22	30	13:00	13:39
	18:30 - 19:30	23.69	69	17:45	18:54	23.3	49	18:05	18:54
M	10:00 - 10:30	27.9	45	9:25	10:10	17.61	36	9:50	10:26
	13:30 - 14:00	20.64	40	12:55	13:35	17.92	39	13:15	13:54
	18:30 - 19:30	26.95	63	17:30	18:33	17.61	36	18:15	18:51
N	10:00 - 10:30	9.47	24	9:55	10:19	9.81	23	10:05	10:28
	13:30 - 14:00	8.29	32	13:25	13:57	7.97	19	13:40	13:59
	18:30 - 19:30	9.47	24	18:55	19:19	8.13	21	18:30	18:51

Table 3: The mean values of travel time for O-D pairs L, M and N

Time-dependent Fastest Path with Multiple Drops

In this section of the paper, we describes how to apply the algorithm for time-dependent fastest path with multiple drops. This can be applied directly to the logistical system where a company has many customers to serve at different locations. For practical reasons, we scope our experiments to a limited number of customers to be served because, in real world, a company usually, has only about 8 working hours per day and the maximum number of customers to be served is normally less than 10. The mathematical programming for this type of problem can be formulated as follow.

$$\min \sum_{i \in N} \sum_{j \in N} t_{ij}^{d_i} x_{ij} \tag{8}$$

S.T.

$$\sum_{j \in N} x_{ij} = 1 \tag{9}$$

$$\sum_{j \in N} x_{ji} - \sum_{j \in N} x_{ij} = 0 \tag{10}$$

$$\left(a_j - a_i + s_i + t_{ij}^{d_i}\right) x_{ij} \geq 0 \tag{11}$$

$x_{ij} \in \{0,1\}$; $\forall(i,j) \in N$
Where
N = set of all nodes in the network, x_{ij} = 1 if arc ij is a part of the shortest path; 0 otherwise, $t_{ij}^{d_i}$ = travel time from node i to node j with staring time from node i equal to d_i, a_i = arrival time at node i, s_i = service time at node i
The objective function (8) is to minimize the total travel time used on the network. Constraints (9) and (10) are the conventional constraints for network problem. Constraint (11) says that the link from node i to node j is selected then the arrival time at node j must be greater than or equal to the sum of arrival time at node i, service time at node i and travel time from node i to node j. We assumed in this work that the service time is equal to 45 minutes.
The experiments with four O-D pairs were conducted. For the first O-D pair, the trip starts at node P1_A then goes to P1_B, P1_C and P1_D, respectively. For the second O-D pair, the trip starts at node P2_A then goes to P2_B, P2_C, P2_D, P2_E and P2_F respectively. For the third O-D pair, the trip starts at node P3_A then goes to P3_B, P3_C, P3_D, P3_E and P3_F respectively. The fourth O-D pair starts from node P4_A then goes to P4_B and P4_C respectively. **Error! Reference source not found.**The resulted time-dependent fastest path based on the travel time data on August 6, 2012 can be illustrated in Table 4. In addition, the travel time between the shortest paths and time-dependent fastest paths on 40 Mondays from the historical data were compared as shown in **Error! Reference source not found.**.

Problem	Path	Distance (km)	Time		
			Start	End	Total
P1	P1_A, P1_D, P1_B, P1_C, P1_A	32.08	14:00	17:26	3:26
P2	P2_A, P2_C, P2_D, P2_F, P2_E, P2_B, P2_A	56.21	9:00	14:17	5:17
P3	P3_A, P3_B, P3_F, P3_C, P3_D, P3_E, P3_A	15.53	9:00	13:33	4:33
P4	P4_A, P4_C, P4_B, P4_A	279	14:00	18:41	4:41

Table 4: The time-dependent fastest paths for all four O-D pairs

Problem	Distance (km)			Time (min)				Best route	
	Shortest	Fastest	Diff	Shortest	Fastest	Diff	%*	Distance	Time
P1	30.5	32.7	2.2	218	200	18	22.61	30.5	198
P2	50.9	54.4	3.5	338	308	30	26.46	51.9	299
P3	14.4	15.8	1.4	281	269	12	21.25	14.4	265
P4	93.0	105.9	12.9	368	301	67	24.35	96.8	268

Table 5: Comparison of the mean travel time of shortest and time-dependent fastest paths

Conclusion
In this paper, the algorithm to find time-dependent fastest path was considered. A main problem is the complexity and computational time of the algorithm when there are many time interval to be considered. The pre-processing and data architecture were proposed to handle this problem. The efficiency of the proposed concepts were examined with the real road network in Bangkok with 3,500 × 3,500 nodes with the time interval of 5 minutes for a day. The results shows that the proposed concepts works quite well and can be practically applied to the real world problem. The time-dependent fastest path with arrival time window and with multiple drops were also examined. The total cost saving between the shortest and time-dependent fastest paths were compared to better understand the travelers 'behaviors. Based on the proposed concepts, the future work could be the development of decision support system for travelers using real-time traffic data and hierarchical optimization for trip planning.

References

Chen, A. and Ji, Z.W. (2005) “Path finding under uncertainty,” Journal of Advanced Transportation, vol. 39 (1), pp. 19 - 37.

Dijkstra, E. W. (1959) "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, pp. 269 - 271.

Fu, L.P. and Riett, L.R. (1998) “Expected shortest paths in dynamic and stochastic traffic network,” Transportation Research Part B-Methodological, vol. 32 (7), pp. 499 - 516.

Hart. E.P., Nilsson, N.J., and Raphael, B. (1968) “A formal basis for the heuristic determination of minimum cost paths,” IEEE Transaction of System Science and Cybernetics, SSC-4 (2), pp. 100 - 107.

Hashimoto, H., Yagiura, M., Ibaraki, T. (2008) “An iterated local search algorithm for the time-dependent vehicle routing problem with time windows,” Discrete optimization 5, pp. 434 -456.

Ichoua, S. Gendreau, M., and Potvin, J.Y. (2003) “Vehicle dispatching with time-dependent travel time,” European Journal of Operational Research, vol. 144, pp. 379 - 396.

Kanoulas, E., Du, Y., Xia, T., and Zhang, D. (2006) “Finding fastest paths on a road network with speed patterns,“ In: Proceedings of the 22nd International Conference on Data Engineering.

Lu, X (2009) "Optimal adaptive departure time choices with real-time traveller information considering arrival time reliability," Master Thesis in Civil Engineering, University of Massachusetts Amherst.

Tilahun, N.Y. and Levinson, D.M. (2010) “A moment of time: reliability in route choice using stated preference,” Journal of Intelligent Transportation System, vol. 14(3), pp. 179 – 187.

Woensel, T.V., Kerbache, L., Peremans, H., and Vandaele, N. (2008) “Vehicle routing with dynamic travel times: a queueing approach,” Discrete optimization, vol. 186, 990 - 1007.

Wu, X. and Nie, Y. (2009) “Implementation issues for the reliable a priori shortest path problem,“ Transportation Research Record, vol. 2091, pp. 51 - 60.