Songklanakarin J. Sci. Technol.
35 (4), 489-499, Jul. - Aug. 2013

SJST SONGKLANAKARIN JOURNAL OF SCIENCE AND TECHNOLOGY
http://www.sjst.psu.ac.th

Original Article

Maximum flow in road networks with speed-dependent capacities – application to Bangkok traffic

Elvin J Moore*, Wisut Kichainukon, and Utomporn Phalavonk

Department of Mathematics, Faculty of Applied Science,
King Mongkut's University of Technology North Bangkok, Bangsue, Bangkok, 10800 Thailand.

Received 11 July 2012; Accepted 2 June 2013

Abstract

A road network can be modeled as a graph with a set of nodes representing intersections and a set of weighted edges representing road segments between intersections. In this paper, a traffic flow problem is studied, where edge weights represent road capacities (maximum vehicles per hour) that are functions of the traffic speed (km/hr) and traffic density (vehicles per kilometer). To estimate road capacities for a given speed, empirical data on safe vehicle separations for a given speed are used. A modified version of the Ford-Fulkerson algorithm is developed to solve maximum flow problems with speed-dependent capacities, with both one-way and two-way flows allowed on edges and with multiple source and target nodes. The modified algorithm is used to estimate maximum traffic flow through a selected network of roads in Bangkok. It was found that the maximum safe traffic flow occurs at a speed of 30 km/hr.

Keywords: maximum traffic flow, flow-dependent capacities, Ford-Fulkerson algorithm, Bangkok roads

1. Introduction

In many cities, traffic jams are a big problem. There have been many attempts to try to improve road design and traffic flow using detailed empirical measurements of traffic flow, analysis of traffic accidents, and redesign of accident "black spots", revision of traffic laws and methods of enforcement, theory of physics, mathematical modeling and computer simulation, among others (e.g., Khisty and Lall, 2003; Roess *et al.*, 2004; Rotwannasin and Choocharukul, 2005; Touya, 2007; Garofalakis *et al.*, 2007; Bonzani and Arikawa, 2007; Miller, 2009; Wikipedia, 2012). It is well known that the traffic flow on a road (vehicles per hour) and capacity (maximum vehicles per hour) of a road depends on both the traffic speed (kilometers per hour) and traffic density (vehicles per kilometer). For example, Bonzani and Arikawa (2007) reported experimental measurements showing how the flow on a road changed as the density of the vehicles (vehicles per kilometer) changed.

In the experimental measurements reported in Bonzani and Arikawa (see Figure 1), the traffic flow along a road increased approximately linearly as the density increased until a critical density u_c was reached. At the critical density, the flow

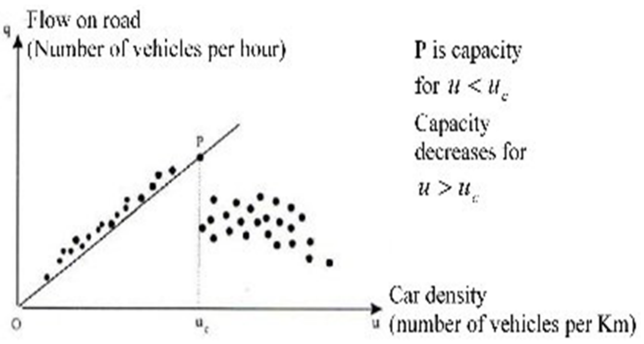


Figure 1. Experimental traffic flow measurements (adapted from Bonzani and Arikawa, 2007).

* Corresponding author.
Email address: ejm@kmutnb.ac.th

on the road was at a maximum and may be regarded as being the maximum capacity of the road. As the density increased past this critical point, there was a big reduction in the capacity of the road and the traffic flow. The road capacity continued to decrease as the density increased and a traffic jam could easily occur. The region below the critical density corresponds to the "free-flow region" where the traffic flow is less than the road capacity, whereas the region above the critical density corresponds to the "congested flow region" in which the flow is limited by the road capacity. It can be seen that there is usually a "low-speed" value of density and a "high-speed" value of density corresponding to a given capacity. The capacity of a road is therefore not a constant, but is a function of traffic speed and traffic density. Because the capacity of each road in a traffic network is flow-dependent, the capacity of the road network must also be flow-dependent. The properties of networks with flow-dependent capacities have been studied by a number of authors (e.g. Shahrokhi and Matula, 1990; Burkhard *et al.*, 1993; Fleischer and Tardos, 1998; Baumann and Köhler, 2007).

The methods of graph theory can be used to study traffic flow through road networks (e.g., Köhler *et al.*, 2002; Bertelle *et al.*, 2003; Hu *et al.*, 2008; Tizghadam and Leon-Garcia, 2009; Halaoui, 2010). The Ford-Fulkerson algorithm is a well-known graph theory method for calculating maximum flow through networks (e.g., Ford and Fulkerson, 1957, 1962; West, 2001; Gross and Yellen, 2006). In its standard form, the algorithm requires that each edge is a directed edge with a fixed capacity. The direction of each edge is required so that "forward" and "backward" directions of flow are uniquely specified in the construction of the *f*-augmenting paths used in the optimization of the flow. In a road network, the roads are usually two-way roads and the capacity is a function of the traffic speed or density. The standard algorithm must therefore be modified so that at each flow augmentation step of the algorithm the direction of flow on a two-way edge can be uniquely specified as a "forward" flow direction or a "backward" flow direction. In this paper, we develop a modified version of the Ford-Fulkerson algorithm, which can be used to calculate the maximum flow through a network with two-way edges and with flow-dependent capacities on the edges. As an example of the application of the modified algorithm, we calculate the maximum traffic flow from a given set of source nodes to a given set of target nodes in a selected network of roads in Bangkok.

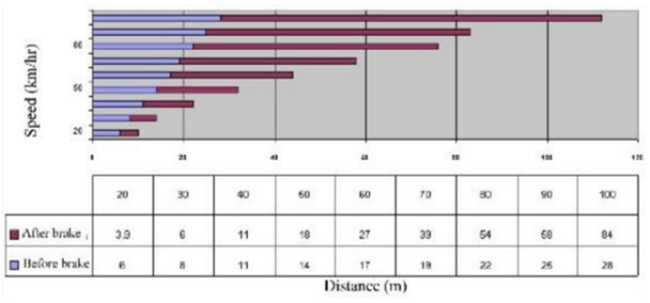
The structure of this paper is as follows. In Section 2, data on safe car separation as a function of speed (Toyota, 2009) are used to estimate road capacity (maximum vehicles per hour) as a function of speed. The Toyota safe car separation data is typical of data contained in highway codes and safe driving manuals in many countries. In Section 3, an algorithm based on the Ford-Fulkerson algorithm is developed, which can solve problems in which edges can be traversed in both directions and in which the capacity can be varied depending on the flow on the edge. In Section 4, an example is given of a network of roads selected from major Bangkok roads. Data available from the Bangkok government and Bangkok traffic police (e.g. Traffic statistics, 2007; Thai traffic, 2007) includes road lengths and average speed values on these roads each day for representative times of a day. These data can be combined with the safe car separation data from Section 2 to estimate safe capacities for each lane of a road in each direction in the selected network. In Section 5, examples are given of the maximum traffic flows computed between selected nodes in the network using a Matlab program based on the algorithm in Section 3. The paper concludes with a discussion.

2. Road Capacities as a Function of Speed

Estimation of road capacities as a function of speed is a difficult problem due to the many different types of vehicles on the roads and the varying driving habits of the vehicle drivers. In this paper, we estimate road capacities as a function of speed by using data on safe car separation distances as a function of speed. A more detailed discussion of the effects of this assumption is given in the discussions section.

Data giving safe stopping distances as a function of speed are shown in Figure 2. The data are adapted from Toyota (2009), but as noted in Section 1, similar data are available in highway codes and safe driving manuals of many countries. The stopping distance includes two distances. The first distance is estimated as the distance that a car will travel in the time that the driver takes to see danger and apply the brake. In the estimated distances in Figure 2, this reaction time is taken as approximately one second. The second distance is the distance that the vehicle will travel after the driver applies the brakes. This second distance can be estimated from testing on actual vehicles or from physics.

Using the safe stopping distances in Figure 2, we can estimate the "safe" capacity of one lane of road as a function of speed as follows: (1) We use a spline fit of the data on stopping distance after braking to construct a smooth curve of stopping distance after braking versus speed. (2) The total stopping distance (front of car to rear of car in front) is reaction distance plus stopping distance after braking. (3) We then take 5 meters as an average length of car to obtain car



	20	30	40	50	60	70	80	90	100
🟥 After brake	3.9	6	11	18	27	39	54	58	84
🟦 Before brake	6	8	11	14	17	19	22	25	28

Figure 2. Safe stopping distances for cars as a function of speed (adapted from Toyota, 2009).

separation (front of car to front of car in front) as a function of speed. (4) The capacity in cars per hour is then C = 1,000 x speed (km/hr)/car separation (meters). A plot of the estimated safe capacity versus speed is shown in Figure 3. The plot shows that the maximum capacity occurs at approximately 30 km/hr with a slow drop above this speed and a very fast drop below this speed. Therefore, if a road is carrying the maximum safe capacity at, for example, 100 km/hr, then an increase in density will cause the safe speed to drop with a resulting increase in maximum capacity. However, if a road is carrying the maximum safe capacity at, for example, 25 km/hr, then an increase in density will cause the safe speed to drop with a rapid decrease in capacity and a traffic jam can easily occur.

3. Modified Ford-Fulkerson Algorithm for 2-way Flow and Flow-dependent Capacities

In the standard Ford-Fulkerson algorithm (e.g., Ford and Fulkerson, 1957, 1962; West, 2001; Gross and Yellen, 2006), it is assumed that a network can be represented by a directed graph with a maximum of one directed edge joining each pair of nodes. It is also assumed that the capacity $C(i, j)$ of directed edge (i, j) from node i to node j is fixed for all pairs of nodes i,j and that the maximum flow from a single source node s to a single target node t is required. A summary of the standard Ford-Fulkerson algorithm is given in Appendix A.

In this section we describe a modified version of the algorithm, which can compute the maximum flow in a network from a specified set S of sources to a specified set T of targets, where $S \cap T = \varnothing$. It is assumed that the edges in the network can be directed or undirected, that the capacities of an edge can be different in the two possible edge directions, and that the capacity of an edge depends on traffic speed or traffic density. We use the notation given in Table 1.

A key step in the Ford-Fulkerson algorithm is the construction of an *f-augmenting path* (see Appendix A) from a given source s to a given target t in a directed graph with given flows and capacities. The usual procedure is first to try to construct a $s-t$ quasi-path (Gross and Yellen, 2006) by using breadth-first search on the underlying undirected graph with all capacities defined as 0 or 1. If no $s-t$ quasi-path exists, then the maximum flow has been found. If a $s-t$ quasi-path exists then the flow can be increased by

Table 1. Notation.

Matrix	Meaning
$V(i,j)$	Average speed from node i to node j (km per hour)
$C(i,j)$	Capacity of edge from i to j (vehicles per hour)
$F(i,j)$	Flow along edge from i to j (vehicles per hour)
$S(i,j)$	Slack on edge from i to j
$A(i,j)$	Action adjacency matrix entry for edge i to j (used to specify state of network, see Table (2)
$T(s, t)$	Total flow from source s to target t

constructing an f-augmenting path from the quasi-path (Gross and Yellen, 2006). In order to compute the slacks (unused capacities) on the edges during the construction of an f-augmenting path, it is necessary to identify every edge in the $s-t$ quasi-path as either a "forward edge" (flow in direction from $s \rightarrow t$) where flow can be increased or a "backward edge" (flow in direction from $t \rightarrow s$) where flow can be decreased. For one-way streets, there is only one possible direction of flow and therefore assigning an edge in a quasi-path as forward or backwards is always easy.

For two-way streets, there are two possible directions of flow and assigning an edge as forward or backwards depends on the direction of flow through the edge. For a two-way edge (i, j), we assign edge directions as follows. Firstly, if the flow from node i to node j is $F(i, j) > 0$ then the flow from node j to node i is defined to be $F(j, i) = 0$ and $i \rightarrow j$ is defined as the direction of the edge. If the flow on a two-way edge between nodes i and j is zero, i.e., $F(i, j) = F(j, i) = 0$, then the direction is initially taken as not defined. However, if a two-way edge with zero flow appears in a $s-t$ quasi-path, then the direction of the edge is defined to be the "forward" $s \rightarrow t$ direction and the flow can then be increased in this forward direction. Note that the direction assigned to a two-way edge in the algorithm can be changed, but only when the flow through it is zero. In our algorithm, we use an "action adjacency matrix" A to specify the state of each edge in the network before each breadth-first search for an $s-t$ quasi-path. In particular, A is used to specify if flow can be increased if an edge is a "forward edge" on the quasi-path or decreased if an edge is a "backward edge" on the quasi-path. The definition of A is given in Table 2.

The main steps in our modified Ford-Fulkerson algorithm are as follows:

1. Initialization:

a) Input an average speed data matrix V for each street for each direction and a length data matrix L for each street. In our examples, we use data for the 7:00-9:00 am morning peak period from Traffic statistics (2007) and Thai traffic (2007) (see Table 7 in Appendix B).

b) Use the safe capacity-speed curve (Figure 3) to construct a *capacity matrix* (C) for the network. Note that the capacity matrix is, in general, not symmetric. For one-way streets, if $C(i, j) \neq 0$ then $C(j, i) = 0$. For two-way streets, if $C(i, j) \neq 0$, then $C(j,i) \neq 0$ also, but in general $C(i, j) \neq C(j,i)$.

c) Select two disjoint sets of nodes as a set of source nodes S and a set of target nodes T. Note that the algorithm can compute the maximum safe capacity flow for any choice of disjoint sets of source and target nodes.

d) If the set of source nodes contains one node, then call this node s. If the set of target nodes contains one node, then call this node t.

e) If the set S of source nodes contains more than one node, add a virtual source node s and directed edges from s to each node in S with each added edge having large capacity and zero length. If the set T of target nodes contains more than one node, add a virtual target node *t* and directed edges from each node in *T* to *t* with each added edge having large capacity and zero length. As for the standard Ford-Fulkerson algorithm, our modified algorithm assumes that the network has one source and one target. The addition of a single virtual source and/or virtual target is a well-known method of solving maximum flow problems with more than one actual source and/or actual target.

f) Set flow matrix $F = 0$ and total network flow from *s* to *t* as $T(s, t) = 0$.

2. Iteration: Repeat the following steps:

a) For the given flow matrix *F* and capacity matrix *C* set an action adjacency matrix *A* as defined in Table 2. This action adjacency matrix defines edge directions for the current iteration step.

b) Set an adjacency matrix *B* for breadth-first search as $B(i,j) = 0$ if $A(i,j) = 0$, and $B(i,j) = 1$ if $A(i,j) \neq 0$.

c) Carry out a breadth-first search (Gross and Yellen, 2006) to find a $s-t$ quasi-path. If no $s-t$ quasi-path exists, then a maximum network flow has been found. Go to Output step 3. Otherwise continue:

d) Construct an *f*-augmenting path:

i) For each edge $i \rightarrow j$ on the quasi-path assign the direction as a forward or backward edge as follows:

A If $A(i,j) = 1$, then the edge is a forward edge. The slack is $S(i,j) = C(i,j) - F(i,j) > 0$.

B If $A(i,j) = 2$, then the edge is a backward edge. The flow is $F(j,i) > 0$.

C If $A(i,j) = 3$, then the direction $i \rightarrow j$ is defined as a forward direction. The slack is $S(i,j) =$, $C(i,j)$ since $F(i,j) = 0$ when $A(i,j) = 3$.

ii) Compute the flow augmentation Δ along the $s-t$ quasi-path, as the minimum of the slacks $S(i,j)$ on the forward edges and the flows $F(j,i)$ on the backward edges.

e) Update the flow matrix *F* on the $s-t$ quasi-path:

i) If (i, j) is a forward edge, then set $F(i,j) = F(i,j) + \Delta$.

ii) If (j, i) is a backward edge, then set $F(i,j) = F(j,i) - \Delta$.

f) Update the total network flow from *s* to *t* as $T(s,t) = T(s,t) + \Delta$. Return to step 2(a).

3. Output:

a) Compute the minimum cut. Let V_N be the set of all nodes in the network. From the final breadth-first search, let V_S be set of nodes in same component as source *s*. Using the final breadth-first search matrix, find the set V_T of nodes in the same component as target *t*. Check that $V_S \cap V_T = \varnothing$ and $V_S \cup V_T = V_N$ (if not, there is an error in the algorithm). Find the partition cut $\langle V_S, V_T \rangle$ of forward edges from any node in V_S to any node in V_T. Then, $\langle V_S, V_T \rangle$ is a minimum edge-cut and the sum of capacities of set of edges $\langle V_S, V_T \rangle$ is the minimum cut.

b) Check that total network flow is a maximum by checking that total network flow is equal the minimum cut (Maximum flow-Minimum cut theorem (Gross and Yellen, 2006)).

Note: A failure of the check indicates an error in the algorithm.

c) Construct a set of possible maximum flow paths from the set of source nodes to the set of target nodes. In general, the set of paths is not unique. We obtain a unique set of paths by using the Dijkstra algorithm (Gross and Yellen, 2006) to generate paths in order of increasing travel time on path.

Table 2. Definition and meaning of action adjacency matrix entries $A(i,j)$ and $A(j,i)$.

Capacity	Flow conditions	A entries	Meaning
$C(i,j) = 0$ $C(j,i) = 0$		$A(i,j) = 0$ $A(j,i) = 0$	Edges $i \rightarrow j$ and $j \rightarrow i$ do not exist in network
$C(i,j) > 0$ $C(j,i) = 0$	$F(i,j) = 0$ $F(j,i) = 0$	$A(i,j) = 1$ $A(j,i) = 0$	Edge $i \rightarrow j$ is included in breadth-first search (BFS). Edge $j \rightarrow i$ is not included in BFS. If $i \rightarrow j$ occurs in (*s*, *t*) quasi path, then $i \rightarrow j$ is "forward edge" and $F(i,j)$ can be increased.
$C(i,j) > 0$ $C(j,i) > 0$	$F(i,j) = 0$ $F(j,i) = 0$	$A(i,j) = 3$ $A(j,i) = 3$	Edges $i \rightarrow j$ and $j \rightarrow i$ are both included in BFS. If an edge occurs in (*s*, *t*) quasi path, then that edge is defined as a "forward edge" and flow can be increased on that edge.
$C(i,j) > 0$ $C(j,i) \geq 0$	$0 < F(i,j) < C(i,j)$ $F(i,j) = 0$	$A(i,j) = 1$ $A(j,i) = 2$	Edges $i \rightarrow j$ and $j \rightarrow i$ are both included in BFS. If $i \rightarrow j$ occurs in (*s*, *t*) quasi path, then (i,j) is a "forward edge" and $F(i,j)$ can be increased. If $j \rightarrow i$ occurs in (*s*, *t*) quasi path, then (i,j) is a "backward edge" and $F(i,j)$ can be decreased.
	$F(i,j) = C(i,j)$ $F(j,i) = 0$	$A(i,j) = 0$ $A(j,i) = 2$	Edge $i \rightarrow j$ is not included in BFS. Edge $j \rightarrow i$ is included in BFS. If $j \rightarrow i$ occurs in (*s*, *t*) quasi path, then (i,j) is a "backward edge" and $F(i,j)$can be decreased.

4. Construction of a Network for Selected Roads in Bangkok

The Traffic and Transportation Department of Bangkok and the Police collect data for traffic flow on Bangkok roads. We used the data for 2007 as an example (Traffic statistics, 2007). These data contain details of speed on the roads, length of roads, number of roads, and period of time. The data gives an average speed on 37 Bangkok roads for in– and outbound directions for the periods 7.00-9.00 a.m. and 4.00-6.00 p.m., where the "inbound" direction is typically the direction of traffic flow for people going to work in the morning and the "outbound' direction is typically the direction of traffic flow for people returning home in the evening. We used this data to select a road network of major Bangkok roads containing 53 nodes and 83 edges. Details of the selected roads and intersections (nodes) on each road are shown in the map in Figure 3 and in the list given in Table 6 in Appendix B. A list of the average speeds and the lengths for the edges in the selected network is given in Table 7 in Appendix B. For a given average speed on each road (Thai traffic, 2007), we estimated the maximum safe capacity of that road using the data shown in Figure 3.

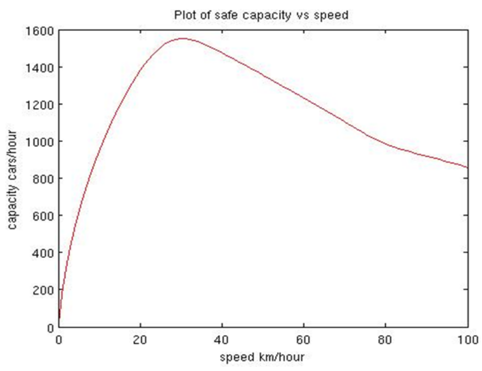


Figure 3. Safe road capacity for cars as a function of speed.

5. Results

An example of the results obtainable from the Matlab program based on the algorithm in Section 3 are shown in Tables 3 to 5 and Figure 5 for maximum flow from source nodes 3,4,7 to target nodes 46,48,49, repectively. The computation time on a PC running at 2.9 GHz was approximately 1.4 seconds. Table 3 shows a possible set of paths for maximum flow. In general, the set of paths is not unique. The set of paths shown in Table 3 was obtained by using the Dijkstra shortest path algorithm (Gross and Yellen, 2006) to construct paths in increasing order of travel time. An alternative procedure would be to use the Dijkstra algorithm to construct paths in decreasing order of maximum flow. Table 4 shows

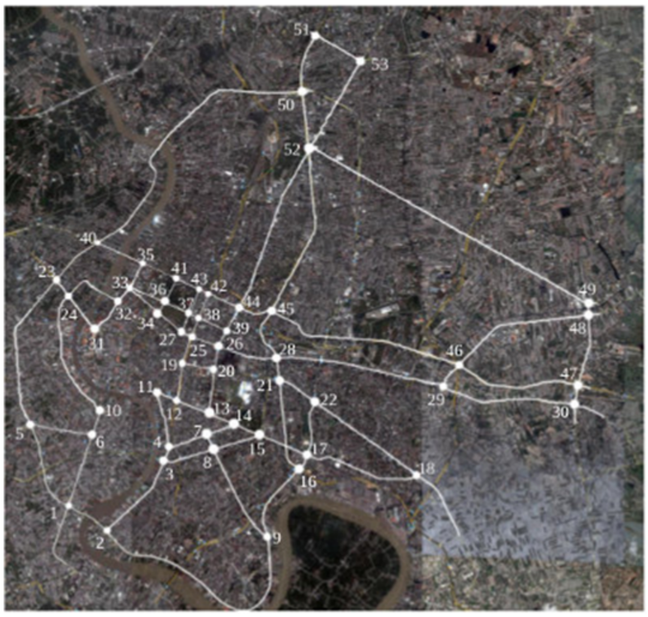


Figure 4. Map showing nodes (intersections) and edges (road segments) in a selected network of Bangkok roads. Details of nodes and edges are listed in Tables 6 and 7.

Table 3. Possible set of paths for maximum flow from node 3,4,7 to node 46,48,49. Paths selected in order of increasing travel time.

Path ID	Time (min)	Flow (vehicles/hr)	Nodes on path
1	35.07	598	4,11,12,19,25,26,28,45,46
2	35.96	116	4,11,12,19,25,26,39,44,45,46
3	37.13	1,184	3,8,9,16,21,28,45,46
4	37.80	368	7,14,13,20,26,39,44,45,46
5	39.62	1,284	7,14,13,20,26,28,29,46
6	43.96	583	4,11,12,13,20,26,28,29,46
7	48.61	598	3,2,9,16,21,28,29,46
8	71.18	760	4,11,12,19,25,26,39,44,52,49
9	93.87	1,753	3,2,1,5,23,40,50,52,49

A minimum cut: (28,29), (45,46), (52,49)
Minimum cut: Flow = 7,244
Maximum flow: Flow = 7,244

Table 4. Flow on edges for maximum flow from source 3,4,7 to target 46,48,49.

Edge	Flow	Capacity	Slack
(1,5)	1753	2258	505
(2,1)	1753	1995	242
(2,9)	598	2401	1803
(3,2)	2351	2351	0
(3,8)	1184	2003	819
(4,11)	2057	2057	0
(5,23)	1753	2527	774
(7,14)	1652	1652	0
(8,9)	1184	2463	1279
(9,16)	1782	2401	619
(11,12)	2057	2307	250
(12,13)	583	2307	1724
(12,19)	1474	2404	930
(13,20)	2235	2235	0
(14,13)	1652	2509	857
(16,21)	1782	1782	0
(19,25)	1474	2404	930
(20,26)	2235	2235	0
(21,28)	1782	1782	0
(23,40)	1753	2527	774
(25,26)	1474	2439	965
(26,28)	2465	2465	0
(26,39)	1244	2235	991
(28,29)	2465	2465	0
(28,45)	1782	2370	588
(29,46)	2465	2493	28
(39,44)	1244	2235	991
(40,50)	1753	2527	774
(44,45)	484	2154	1670
(44,52)	760	2527	1767
(45,46)	2266	2266	0
(50,52)	1753	2370	617
(52,49)	2513	2513	0

Minimum cut: (28,29),(45,46),(52,49)
Flow: 7,244
Units of flow, capacity and slack are: Vehicles per hour

the capacities, maximum flows and slack (unused capacity) on edges for maximum flow from sources 3,4,7 to targets 46,48,49, respectively. Table 5 shows the capacities, maximum flows and slacks for the nodes on the maximum flow paths. Figure 5 shows a graphic of the nodes, edges and flows for the maximum flow paths.

6. Discussion and Conclusions

The results show that the maximum safe flow of a road network occurs when all vehicles maintain the same speed of approximately 30 km/hr. At speeds above this value, the maximum flow increases as speed decreases and the traffic flow should be reasonably smooth. Below 30 km/hr the maximum flow decreases rapidly as speed decreases and traffic jams can easily occur.

The maximum flow paths calculated in this paper could be regarded as upper bounds that might be obtainable under expressway conditions with all drivers maintaining constant speed, a "safe" separation distance and with no–lane changing. There are clearly many factors that we have not considered in this paper that would change the capacity of a road. Some of these factors are as follows. In calculating the capacities for the edges we have not included the effects of traffic lights and intersections. These effects could be included by reducing the capacities of incoming edges to a node to allow for the time stopped at an intersection. We have also not included the effects of cars moving at different speeds, and changing lanes. We have also not included the fact that road capacities at a given speed could be increased if drivers "tail-gated', i.e., maintained less than the recommended safe separation distances. To estimate capacities with these effects included would require a stochastic treatment that is beyond the scope of this paper. All of these neglected effects would be expected to both decrease the road capacities and greatly increase the risk of accidents. However, it should be noted that the algorithms developed in this paper could rapidly find the maximum flow through a road network if good estimates of the capacities of the roads in the network were available.

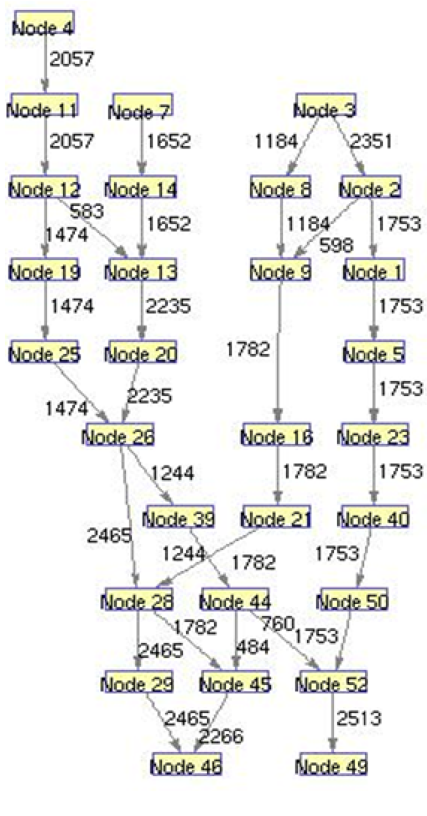


Figure 5. Plot of maximum flow paths from node 3,4,7 to node 46,48,49. Node 48 is not reached for maximum flow.

Table 5. Flow through nodes for maximum flow from node. 3,4,7 to node 46,48,49.

Node	Capacity		Flow		Slack	
	In	Out	In	Out	In	Out
1	7033	7089	1753	1753	5280	5336
2	6747	6611	2351	2351	4396	4260
3	6853	6569	0	3535	6853	3034
4	6259	6060	0	2057	6259	4003
5	7278	7373	1753	1753	5525	5620
6	7298	7460	0	0	7298	7460
7	5378	6092	0	1652	5378	4440
8	8502	8502	1184	1184	7318	7318
9	6859	6145	1782	1782	5077	4363
10	2430	4889	0	0	2430	4889
11	4566	4374	2057	2057	2509	2317
12	7152	7220	2057	2057	5095	5163
13	6988	7051	2235	2235	4753	4816
14	6468	6793	1652	1652	4816	5141
15	6819	7103	0	0	6819	7103
16	6766	6178	1782	1782	4984	4396
17	9725	9069	0	0	9725	9069
18	4225	4609	0	0	4225	4609
19	6836	6977	1474	1474	5362	5503
20	6644	6770	2235	2235	4409	4535
21	6252	6070	1782	1782	4470	4288
22	6276	6526	0	0	6276	6526
23	7139	6663	1753	1753	5386	4910
24	8591	6067	0	0	8591	6067
25	9251	9064	1474	1474	7777	7590
26	6846	8941	3709	3709	3137	5232
27	6688	6799	0	0	6688	6799
28	8686	7205	4247	4247	4439	2958
29	7064	7027	2465	2465	4599	4562
30	4974	4523	0	0	4974	4523
31	4041	2092	0	0	4041	2092
32	4784	7471	0	0	4784	7471
33	9259	9467	0	0	9259	9467
34	7031	7078	0	0	7031	7078
35	6898	6801	0	0	6898	6801
36	9195	9195	0	0	9195	9195
37	8561	8561	0	0	8561	8561
38	8990	8990	0	0	8990	8990
39	6497	6567	1244	1244	5253	5323
40	7461	7009	1753	1753	5708	5256
41	6872	6825	0	0	6872	6825
42	6780	7164	0	0	6780	7164
43	6581	6482	0	0	6581	6482
44	9514	9267	1244	1244	8270	8023
45	9424	8947	2266	2266	7158	6681
46	9803	10061	4731	0	5072	10061
47	7471	7477	0	0	7471	7477
48	7401	7548	0	0	7401	7548
49	5022	4963	2513	0	2509	4963
50	6679	6672	1753	1753	4926	4919
51	4311	4775	0	0	4311	4775
52	11671	11528	2513	2513	9158	9015
53	4932	5012	0	0	4932	5012

Flow out from sources 3,4,7 = 7244 Flow into targets 46,48,49 = 7244
Units of capacity, flow and slack are: Vehicles per hour

The results in this paper used data from 2007. The methods developed could be applied to other years as the data is available from the Traffic and Transportation Department of Bangkok and the Bangkok traffic police. The algorithm and Matlab computer program that we have developed rapidly solved the 53 node and 83 edge network that we studied and they are capable of solving larger size problems. We expect that the algorithm and programs can be applied to any type of network where edges may have two directions and the capacities are flow-dependent.

References

Baumann, N. and Köhler, E. 2007. Approximating earliest arrival flows with flow-dependent transit times. Discrete Applied Mathematics. 155, 161-171.

Bertelle, C., Dutot, A., Lerebourg, S. and Olivier, D. 2003. Road traffic management based on ant system and regulation model, Proceedings of the International Workshop on Modeling and Applied Simulation, Italy, 35-43.

Bonzani, I. and Arikawa, S. 2007. Hyperbolicity Analysis of a Class of Dynamical Systems modeling Traffic. Applied Mathematics Letters. 20, 933-937.

Burkhard, R.E., Dlaska, K. and Klinz, B. 1993. The Quickest Flow Problem. Mathematical Methods of Operations Research. 37, 31-58.

Fleischer, L. and Tardos, E. 1998. Efficient continuous-time dynamic network flow algorithms. Operations Research Letters. 23, 71-80.

Ford, L.R.Jr. and Fulkerson, D.R. 1957. A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem. Canadian Journal of Mathematics. 9, 210-218.

Ford, L.R.Jr. and Fulkerson, D.R. 1962. Flows in Networks. Princeton University Press, Princeton, New Jersey, U.S.A.

Garofalakis, J., Polyxeni, N. and Athanasios, P. 2007. Vehicle Routing and Road Traffic Simulation, A Smart Navigation System. In Proceedings of 11th Panhellenic Conference on Informatics (PCI2007), Patras, Greece, 611-623.

Gross, J.L. and Yellen, J. 2006. Graph theory and its applications. 2nd Edition. Chapman & Hall/CRC Boca Raton, New York, U.S.A.

Halaoui, H.F. 2010. Intelligent Traffic System: Road Networks with Time-Weighted Graphs. International Journal for Infonomics. 3, 350-359.

Hu, M.-B., Jiang, R., Wu, Y.-H., Wang, W.-X., and Wu, Q.-S. 2008. Urban traffic from the perspective of dual graph. The European Physical Journal. B 63, 127-133.

Khisty, C.J. and Lall, B.K. 2003. Transportation Engineering: An Introduction. 3rd Edition. Prentice-Hall, Inc., New Jersey, U.S.A.

Köhler, E., Langkau, K. and Skutella, M. 2002. Time-Expanded Graphs for Flow-Dependent Transit Times. Lecture Notes in Computer Science, Algorithms – ESA2002. 2461, 49-56.

Miller, J. 2009. Determining Maximum Vehicular Flow for City Evacuation Planning. REU Program 2009, Jackson State University, Mississippi, U.S.A.

Roess, R.P., Prassas, E.S. and McShane, W.R. 2004. Traffic Engineering, 3rd Edition. Pearson Education International, Upper Saddle River, New Jersey, U.S.A.

Rotwannasin, P. and Choocharukul, K. 2005. Transferability of HCM to Asian Countries: An Exploratory Evidence From Bangkok's Multilane Highways. In Proceedings of the 3rd International SIIV Congress, Bari, Italy, September 2005.

Shahrokhi, F. and Matula, D.W. 1990. The Maximum Concurrent Flow Problem, Journal of the Association for Computing Machinery. 37, 318-334.

Thai traffic 2007. http://www.trafficpolice.go.th/index.php [May 31, 2012].

Tizghadam, A. and Leon-Garcia, A. 2009. A Graph Theoretical Approach to Traffic Engineering and Network Control Problem, Proceedings of the 21st International Teletraffic Congress (ITC21), Paris, France, 1-8.

Touya, G. 2007. A Road Network Selection Process Based on Data Enrichment and Structure Detection. Transactions in GIS. 14, 595-614.

Toyota, 2009. Road Safety by Toyota. http://www.geocities.com/nida4phuket/drivefast.htm [July 15, 2009]. Note: The geocities.com site does not exist anymore. [May 31, 2012]. Similar data is, however, available from Highway Code and Safe Driving manuals supplied to drivers in many countries.

Traffic statistics 2007. http://www.bangkok.go.th/traffic [May 31, 2012].

West, D.B. 2001. Introduction to Graph Theory. 2nd Edition. Prentice Hall, New Jersey. U.S.A.

Wikipedia 2012. Fundamental diagram of traffic flow. http://en.wikipedia.org/wiki/Fundamental_diagram_of_traffic_flow [May 31, 2012].

Appendix A

Standard Ford-Fulkerson algorithm (e.g. Gross and Yellen, 2006)

The algorithm computes maximum flow from source node s to target node t in a directed network with capacities as edge weights.

1) Initialization:

a) Input capacity adjacency matrix C for directed network (all edges are directed edges, edge weights are capacities, $C(i,j) > 0$ gives direction of edge).

b) Input source node s and target node t.

c) Set flow $F(i,j) = 0$ for all edges. Set slack $S(i,j) = C(i,j)$ for all edges. Set total flow from s to t as $T(s,t) = 0$.

2) Iteration: Repeat the following steps:

a) Set an adjacency matrix B for the breadth-first search step as $B(i, j) = 1$ if $S(i,j) > 0$ or $F(j,i) > 0$. Otherwise set $B(i,j) = 0$.

b) Carry out a breadth-first search (Gross and Yellen, 2006) to find a $s-t$ quasi-path. If no $s-t$ quasi-path exists, then a maximum network flow has been found. Go to Output step 3. Otherwise continue:

c) Construct an f-augmenting path:

i. For each edge $i \to j$ on the quasi-path assign the edge direction as a "forward edge" if $S(i,j) > 0$ or a "backward edge" if $F(j,i) > 0$.

ii. Compute the flow augmentation Δ along the $s-t$ quasi-path as the minimum of the slacks $S(i, j)$ on the forward edges and the flows $F(j, i)$ on the backward edges.

d) Update the flow matrix F on the $s-t$ quasi-path:

i) If (i, j) is a forward edge, then set $F(i,j) = F(i,j) + \Delta$.

ii) If (i, j) is a backward edge, then set $F(j,i) = F(j,i) - \Delta$.

e) Update the total network flow from s to t as $T(s,t) = T(s,t) + \Delta$. Return to step 2(a).

3) Output: Maximum flow $T(s, t)$ from source s to target t. A set of maximum flow paths.

Appendix B
Tables of data on selected network of Bangkok roads

Table 6. Nodes and incident roads on selected Bangkok road network (adapted from Thai traffic 2007; Traffic statistics 2007).

Node	Incident Roads	Node	Incident Roads	Node	Incident Roads
1	RatchadaPisek (Thonburi), Somdejphachaotaksin Rama 3	19	Rama 1, Rama 6	37	Sri Ayutthaya, Sawankhalok
2	CharoenKrung (end) Rama 3	20	Phayatai, Rama 1	38	Rama 6, Sri Ayutthaya
3	CharoenKrung (end), Satorn	21	VibhavadiRangsit, Sukhumvit	39	Phayatai, Sri Ayutthaya
4	CharoenKrung (start), CharoenKrung (end), Silom	22	Ratchada Pisek, Sukhumvit	40	Jaransanitwong, Ratchawithee
5	Jaransanitwong, RatchadaPisek (Thonburi), Phetchkasem	23	Jaransanitwong, Borom Rachachonnanee	41	Rama 5, Ratchawithee
6	Prachathipok, Somdejphachaotaksin, Phetchkasem	24	Borom Rachachonnanee Arun Ammarin	42	Rama 6, Ratchawithee
7	Narathiwat Ratchanakarin, Silom	25	Rama 6, Phitsanulok, Petchburi	43	Ratchawithee, Sawankhalok.
8	Narathiwat Ratchanakarin, Satorn	26	Phayatai, Petchburi	44	Phayatai, Phahonyothin, Ratchawithee, Asokedindaeng
9	NarathiwatRatchanakarin, Rama 3	27	Phitsanulok, Sawankhalok	45	Rama 9, VibhavadiRangsit, Asokedindaeng
10	Prachathipok, ArunAmmarin	28	Petchburi, Vibhavadi Rangsit	46	Rama 9, Pattanakarn, Ramkamhaeng
11	Charoen Krung (start), Rama 4	29	Petchburi, Ramkamhaeng	47	Pattanakarn, Srinakarintra
12	Rama 4, Rama 6	30	Petchburi, Srinakarintra	48	Ramkamhaeng, Srinakarintra
13	Phayatai, Rama 4	31	Borom Rachachonnanee Samsen	49	Ladprao, Srinakarintra
14	Rama 4, Silom	32	Samsen, ArunAmmarin	50	Jaransanitwong, Vibhavadi Rangsit
15	Rama 4, Satorn	33	Phitsanulok, SriAyutthaya, Samsen	51	Ngarm WongWan, Vibhavadi Rangsit
16	Rama 3, VibhavadiRangsit	34	Rama 5, Phitsanulok.	52	Phahonyothin, Ladprao, VibhavadiRangsit
17	Rama 3, Rama 4, RatchadaPisek	35	Ratchawithee, Samsen	53	Ngarm Wong Wan, Phahonyothin

Table 7. Average speed values on edges and lengths of edges in network (adapted from Thai traffic 2007; Traffic statistics 2007; 7:00-9:00 a.m. morning peak).

Edge ID	Edge nodes (*i, j*)	Speed (*i, j*) (km/hr)	Speed (*j, i*) (km/hr)	Length (km)	Edge ID	Edge nodes (*i, j*)	Speed (*i, j*) (km/hr)	Speed (*j, i*) (km/hr)	Length (km)
1	(1,2)	35.06	51.82	1.40	43	(26,28)	20.10	0.00	2.00
2	(1,5)	41.15	28.94	2.00	44	(26,39)	15.22	14.36	0.50
3	(1,6)	19.01	25.45	2.40	45	(27,34)	14.10	19.00	0.80
4	(2,3)	14.93	17.16	3.00	46	(27,37)	14.80	13.55	0.30
5	(2,9)	35.06	51.82	9.00	47	(28,29)	20.10	13.13	4.50
6	(3,4)	14.93	17.16	0.50	48	(28,45)	36.46	36.46	2.00
7	(3,8)	12.43	16.01	1.50	49	(29,30)	20.20	13.13	3.90
8	(4,7)	9.40	12.17	1.20	50	(29,46)	30.09	25.36	0.70
9	(4,11)	13.00	13.11	1.60	51	(30,47)	19.72	22.22	0.70
10	(5,6)	37.86	29.53	2.00	52	(31,32)	0.00	18,86	1.20
11	(5,23)	24.13	27.68	5.00	53	(32,33)	23.29	18,86	0.50
12	(6,10)	19,01	25.45	0.80	54	(33,34)	19.00	14.10	0.30
13	(7,8)	20.03	10.15	0.50	55	(33,35)	23.29	18.86	0.90
14	(7,14)	9.40	12.17	0.90	56	(33,36)	13.27	14.20	0.30
15	(8,9)	20.03	10.15	3.30	57	(34,36)	21.45	19.56	0.10
16	(8,15)	12.43	16.01	1.50	58	(35,40)	28.58	12.03	1.50
17	(9,16)	35.06	51.82	2.50	59	(35,41)	12.03	18.58	0.30
18	(10,24)	36.94	0.00	4.50	60	(36,37)	13.37	14.20	0.80
19	(11,12)	16.35	22.19	0.30	61	(36,41)	21.45	19.56	0.80
20	(12,13)	16.35	22.19	1.10	62	(37,38)	13.37	14.20	0.90
21	(12,19)	18.33	16.87	1.20	63	(37,43)	14.80	13.55	0.80
22	(13,14)	16.35	22.19	0.90	64	(38,39)	13.37	14.20	1.30
23	(13,20)	15.22	14.36	1.90	65	(38,42)	18.33	16.87	0.80
24	(14,15)	16.35	22.19	0.90	66	(39,44)	15.22	14.36	1.00
25	(15,17)	16.35	22.19	1.50	67	(40,50)	24.13	27.68	10.50
26	(16,17)	35.06	51.82	0.50	68	(41,43)	12.03	18.58	0.80
27	(16,21)	61.43	36.46	3.00	69	(42,43)	18.58	12.03	1.20
28	(17,18)	16.35	22.19	2.60	70	(42,44)	18.58	18.58	1.90
29	(17,22)	41.15	28.94	1.90	71	(44,45)	14.13	26.22	1.00
30	(18,22)	13.48	11.61	4.20	72	(44,52)	24.01	16.87	6.50
31	(19,20)	15.25	17.41	1.00	73	(45,46)	40.82	25.37	6.00
32	(19,25)	18.33	13.13	0.90	74	(45,52)	61.43	36.46	6.40
33	(20,26)	15.22	14.36	0.80	75	(46,47)	28.94	22.57	4.20
34	(21,22)	11.61	13.48	1.30	76	(46,48)	30.09	25.36	3.90
35	(21,28)	61.43	36.46	0.70	77	(47,48)	19.72	22.22	2.40
36	(23,24)	70.11	47.82	0.60	78	(48,49)	22.22	19.72	0.30
37	(23,40)	24.13	27.68	2.00	79	(49,52)	22.22	22.47	12.00
38	(24,31)	70.11	47.82	1.50	80	(50,51)	61.43	36.46	1.80
39	(24,32)	36.94	23.61	1.90	81	(50,52)	36.46	61.43	2.00
40	(25,26)	33.23	13.13	0.90	82	(51,53)	34.90	24.53	1.80
41	(25,27)	14.10	33.10	2.00	83	(52,53)	24.01	20.84	1.70
42	(25,38)	18.33	16.87	0.60					