Extended network and algorithm finding maximal flows

Tran Ngoc Viet[1], Le Hong Dung[2]
[1]Faculty of Information Technology, Van Lang University, HCM City, Vietnam
[2]Faculty of Information Technology, Central Transport College V, Danang City, Vietnam

Article Info	ABSTRACT
	Graph is a powerful mathematical tool applied in many fields as transportation, communication, informatics, economy, in ordinary graph the weights of edges and vertexes are considered independently where the length of a path is the sum of weights of the edges and the vertexes on this path. However, in many practical problems, weights at a vertex are not the same for all paths passing this vertex, but depend on coming and leaving edges. The paper develops a model of extended network that can be applied to modelling many practical problems more exactly and effectively. The main contribution of this paper is algorithm finding maximal flows on extended networks.

Corresponding Author:

Tran Ngoc Viet,
Faculty of Information Technology,
Van Lang University,
80/68 Duong Quang Ham Street,
Ward 5, Go Vap District, Ho Chi Minh City, Vietnam.
Email: tranngocviet@vanlanguni.edu.vn

1. INTRODUCTION

Graph is a powerful mathematical tool applied in many fields as transportation, communication, informatics, economy, in ordinary graph the weights of edges and vertexes are considered independently where the length of a path is simply the sum of weights of the edges and the vertexes on this path. However, in many practical problems, weights at a vertex are not the same for all paths passing this vertex, but depend on coming and leaving edges. Therefore, a more general type of weighted graphs, called extended weighted graph, is defined in this work. The paper develops a model of extended network that can be applied to modelling many practical problems more exactly and effectively. Based on the results of the study of the problem regarding finding the maximum flow [1, 4, 6-8, 10-18, 20-21, 23-24] and extended graphs [2-3, 5, 9, 19, 22, 25-26] the main contribution of this paper is the revised Ford-Fulkerson algorithm finding maximal flows on extended networks.

2. EXTENDED NETWORK

A network is graph of the traffic G = (V, E), circles V and roads E. Roads can be classified as either direction or non-direction. There are many sorts of means of transportation on the network. The non-direction shows two-way roads while the direction shows one-way roads. Given a group of the functions on the network as follows:

The function of the route circulation possibility $c_E: E \rightarrow R^*, c_E(e)$ the route circulation possibility $e \in E$.
The function of the circle circulation possibility $c_V: V \rightarrow R^*, c_V(u)$ the circle circulation possibility $u \in V$.

$$G = (V, E, c_E, c_V): \text{extended network} \qquad (1)$$

3. FLOW OF THE EXTENDED NETWORK

Given an extended network $G = (V, E, c_E, c_V)$ a source point s and a sink point t.
Set $\{ f(x, y) \mid (x, y) \in E \}$ is called the flow of network G if the requirements are met:

a. $0 \leq f(x, y) \leq c_E(x, y) \ \forall (x, y) \in E$

b.   Any value of point r is referring to neither a sourse point nor a sink point

$$\sum_{(v,r)\in E} f(v, r) = \sum_{(r,v)\in E} f(r, v) \tag{2}$$

c.   Any value of point r is referring to neither a sourse point nor a sink point

$$\sum_{(v,r)\in E} f(v, r) \leq c_V(r) \tag{3}$$

Expression: $v(F) = \sum_{(a,v)\in E} f(a, v)$, is called the value of flow F $\tag{4}$

3.1. The maximum problem

Given an extended network $G = (V, E, c_E, c_V)$, a source point s and a sink point t. The task required by the problem is finding the flow which has a maximum value. The flow value is limited by the total amount of the circulation possibility on the roads starting from source points. As a result of this, there could be a confirmation on the following theorem.

3.2. Theorem 1

Given an extended network $G = (V, E, c_E, c_V)$, a source point s and a sink point t, then exist is the maximal flow [1].

4. THE ALGORITHM FINDING MAXIMAL FLOWS

4.1. Source toward sink algorithm

Input: Given an extended network $G = (V, E, c_E, c_V)$, a source point s and a sink point t. The points in graph G are arranged in a certain order [3].
Output: Maximal flow $F = \{f(x, y) \mid (x, y) \in E\}$.
(1) Start:
The departure flow: $f(x, y) := 0, \forall (x, y) \in E$.
Points from the source points will gradually be labelled L1 for the first time including 5 components:
Form forward label

$L_1(v) = [\uparrow, prev_1(v), c_1(v), d_1(v), bit_1(v)]$ and can be labelled ($\uparrow$) for the second time
$L_2(v) = [\uparrow, prev_2(v), c_2(v), d_2(v), bit_2(v)]$.

Put labeling ($\uparrow$) for source point

$$L_1(s) = [\uparrow, \emptyset, \infty, \infty, 1]$$

The set S comprises the points which have already been labelled ($\uparrow$) but are not used to label ($\uparrow$), S' is the point set labelled ($\uparrow$) based on the points of the set S.
Begin

$$S := \{s\}, \ S' := \emptyset$$

(2) Forward label generate:
(2.1) Choose forward label point:
$Case \ S \neq \emptyset$: Choose the point $u \in S$ of a minimum value. Remove the u from the set S,
$S := S \setminus \{u\}$. Assuming that the forward label of u is $[\uparrow, prev_i(u), c_i(s), d_i(s), bit_i(s)], i = 1 \ or \ 2$.
A is the set of the points which are not forward label time and adjacent to the forward label point u.
Step (2.2).

$Case\ S = \emptyset\ and\ S' \neq \emptyset: Assign\ S := S',\ S' := \emptyset$. Return to step (2.1).
$Case\ S = \emptyset\ and\ S' = \emptyset$: The flow F is the maximum. End.

(2.2.) Forward label the points which are not forward label and are adjacent to the forward label points

Case $= \emptyset$: Return to Step (2.1).

Case $\neq \emptyset$: Choose $v \in A$ of a minimum value. Remove the v from the set $A, A := A \setminus \{v\}$.

Assign forward labeled point *v*:

If $(u,v) \in E, f(u,v)\ < c_E(u,v), bit_i(u) = 1$ put forward label point *v*:

$prev_j(v) := u;$
$c_j(v) := min\{c_i(u), c_E(u,v) - f(u,v)\}, if\ d_i(u) = 0,$
$c_j(v) := min\{c_i(u), c_E(u,v) - f(u,v), d_i(u)\}, if\ d_i(u) > 0;$
$d_j(v) := c_V(v) - \sum_{(i,v)\in E} f(i,v);$
$bit_j(v) := 1, if\ d_j(v) > 0,$
$bit_j(v) := 0, if\ d_j(v) = 0.$

If $(v,u) \in E,\ f(v,u) > 0,$ put forward label point *v*:

$prev_j(v) := u;$
$c_j(v) := min\{c_i(u), f(v,u)\},$
$d_j(v) := c_V(v) - \sum_{(i,v)\in E} f(i,v);\ bit_j(v) := 1.$

If v is not forward label, then return to Step (2.2).

If v is forward label and v is backward label, then making adjustments in increase of the flow. Step (2.3).

If v is forward label and v is not backward label, then add v to *S'*,
$S' := S' \cup \{v\}$, and return to Step (2.2).

(3) Making adjustments in increase of the flow:

Suppose *s* is forward label $[\uparrow, prev_i(s), c_i(s), d_i(s), bit_i(s)]$:

(3.1) Adjustment made from v back to s according to forward label

(3.1.1) Start

$y := v, x := prev_1(v), \delta := c_1(v).$

(3.1.2.) Making adjustments

(i) Case (x, y) the road section whose direction runs from x to *y*:
put $f(x,y) := f(x,y) + \delta..$

(ii) Case (x, y) the road section whose direction runs from y to *x*:
put $f(y,x) := f(y,x) - \delta..$

(iii) Case (x, y) non-direction roads:
If $f(x,y) \geq 0\ and\ f(y,x) = 0$, then
put $f(x,y) := f(x,y) + \delta.$
If $f(y,x) > 0$, then
put $f(y,x) := f(y,x) - \delta.$

(3.1.3) Moving

(i) Case $x = s..$ Step (3.2).

(ii) $x \neq s, put\ x := y\ and\ y := k, k$ is the second component of the forward labeled point x. Then return to Step (3.1.2).

(3.2.) Remove all the labels of the network points, except for the source point *s*. Return to Step (2).

4.2. Sink toward source algorithm

Input: Given an extended network $G = (V, E, c_E, c_V)$, a source point s and a sink point t. The points in graph G are arranged in a certain order [3].

Output: Maximal flow $F = \{f(x, y) \mid (x, y) \in E\}$.

(1) Start:

The departure flow: $f(x, y) := 0, \forall (x, y) \in E$.

Points from the sink points will gradually be labelled L_1 for the first time including 5 components:

Form backward label

$L1(v) = [\downarrow, prev_1(v), c_1(v), d_1(v), bit_1(v)]$ and can be label $(\downarrow)$ for the second time

$L2(v) = (v) = [\downarrow, prev_2(v), c_2(v), d_2(v), bit_2(v)]$.

Put labeling $(\downarrow)$ for sink point:

$$L_1(t) = [\downarrow, \phi, \infty, \infty, 1]$$ the set T comprises the points which have already been labelled $(\downarrow)$ but are not used to label $(\downarrow)$, T' is the point set labelled $(\downarrow)$ based on the points of the set T. Begin

(2) Backward label generate:

(2.1) Choose backward label point:

Case $T \neq \phi$: Choose the point $v \in T$ of a minimum value. Remove the v from the set T, $T := T \setminus \{v\}$. Assuming that the backward label of v is

$$[\downarrow, prev_i(v), c_i(t), d_i(t), bit_i(t)], i = 1 \ or \ 2.$$

B is the set of the points which are not backward label time and adjacent to the backward label point v. Step (2.2).

$Case \ T = \phi \ and \ T' \neq \phi : Assign \ T := T', \ T' := \phi$. Return to step (2.1).

$Case \ T = \emptyset \ and \ T' = \emptyset$: The flow F is the maximum. End.

(2.2) Backward label the points which are not backward label and are adjacent to the backward label points v

Case $B = \phi$: Return to Step (2.1).

Case $B \neq \phi$: Choose $t \in B$ of a minimum value. Remove the t from the set B, $B := B \setminus \{t\}$.

Assign backward labeled point t:

If $(t, v) \in E, f(t, v) < c_E(t, v), bit_i(v) = 1$ put backward label point t:

$prev\_j (t) := v;$
$prev_j(t) := v;$
$c_j(t) := min\{c_i(v), c_E(t, v) - f(t, v)\}, if \ d_i(v) = 0,$
$c_j(t) := min\{c_i(v), c_E(t, v) - f(t, v), d_i(v)\}, if \ d_i(v) > 0;$
$d_j(t) := c_V(t) - \sum_{(i,t) \in E} f(i, t).$
$bit_j(t) := 1, if \ d_j(t) > 0,$
$bit_j(t) := 0, if \ d_j(t) = 0.$

If $(v, t) \in E, \ f(v, t) > 0$, put backward label point t:

$prev_j(t) := v;$
$c_j(t) := min\{c_i(v), f(v, t)\},$
$d_j(t) := c_V(t) - \sum_{(i,t) \in E} f(i, t) ; bit_j(t) := 1.$

If t is not backward label, then return to Step (2.2).

If t is backward label and t is forward label, then making adjustments in increase of the flow. Step (3).

       If t is backward label and t is not forward label, then add t to T',
$T':= T' \cup \{t\}$, and return to Step (2.2).

(3) Making adjustments in increase of the flow:

Suppose t is backward label $[\downarrow, prev_i(t), c_i(t), d_i(t), bit_i(t)]$
(3.1) Adjustment made from v to t according to backward label
(3.1.1) Start

$$x := v, y := prec_1(v), \delta := c_1(v).$$

(3.1.2) Making adjustments
       (i) Case (x, y) the road section whose direction runs from x to y:
put $f(x, y) := f(x, y) + \delta$.
       (ii) Case (y, x) the road section whose direction runs from y to x:
put $f(y, x) := f(y, x) - \delta$.
       (iii) Case (x, y) non-direction roads:
       If $f(x, y) \geq 0$ and $f(y, x) = 0$ then
put $f(x, y) := f(x, y) + \delta$.
       If $f(y, x) > 0$ then
put $f(y, x) := f(y, x) - \delta$.

(3.1.3) Moving
       (i) Case $x = t$. Step (3.2).
       (ii) Case $x \neq t, put$ $x := y$ and $y := k, k$ is the second component of the backward labeled point x.
Then return to step (3.1.2).

(3.2) Remove all the labels of the network points, except for the sink point t. Return to Step (2).

4.3. Theorem 2
       If the value of the route circulation possibility and the circle circulation possibility are integers, then after a limited number of steps, the processing of the maximum network problem will end.
Proof
       According to theorem 1, after each time of making adjustment of the flow, the flow will be increased with certain units (due to c_E is a whole number, c_V is a whole number, and δ is, therefore, a positive whole number). On the other hand, the value of the flow is limited above by the total amount of the circulation possibility at roads leaving the source points. So, after a limited number of steps, the processing of the maximum network problem will end.

4.4. Theorem 3
       Given an $F = \{f(x, y) \mid (x, y) \in E\}$ is the flow on extended network G, a source point s and a sink point t:

$$\sum_{(s,x)\in E} f(s, x) = \sum_{(x,t)\in E} f(x, t) \qquad (5)$$

Proof
       The points of the set V. If x, y is not previous, assign $f(x, y) = 0$

$$\sum_{y\in V}\sum_{x\in V} f(x, y) = \sum_{y\in V}\sum_{x\in V} f(y, x)$$

$$\Leftrightarrow \sum_{y\in V}\left(\sum_{x\in V} f(x, y) - \sum_{x\in V} f(y, x)\right) = 0$$

$$- \sum_{(s,x)\in E} f(s, x) + \sum_{(x,t)\in E} f(x, t) = 0 \qquad \Leftrightarrow \sum_{(s,x)\in E} f(s, x) = \sum_{(x,t)\in E} f(x, t)$$

4.5. The complexity of the algorithm

It is assumed that the road circulation possibility and the point circulation possibility are whole integer. After each round step, to find the roads to increase the amount of circulation on the flow, we have to approve to pass |E|roads in maximum, and in order to adjust the flow we have to approve to pass 2.|V| roads, in maximum.

As a result, the complexity of each time of increasing the flow is $O(|E| + 2.|V|)$.

Mark v* is the value of the maximum flow. So the complexity of the algorithm is

$$O(v^*(|E| + 2.|V|)) \tag{6}$$

5. RESULT OF THE EXPERIMENT

5.1. Example 1

Given an extended network graph Figure 1. The network has six circles, six direction roads and three non-direction roads. The road circulation possibility c_E and the circle circulation possibility c_V. The source point is l, the sink point is 6.
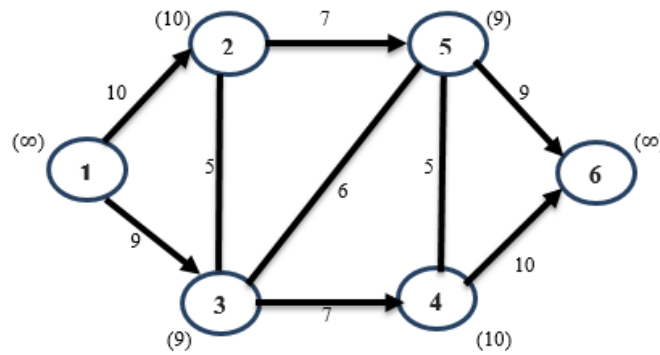

Figure 1. Extended network

Result of the first forward label:
Source point is **1**: forward label [↑, ∅, ∞, ∞, 1]
Point **2**: forward label [↑, 1, 10, 10, 1]
Point **3**: forward label [↑, 1, 9, 9, 1]
Point **4**: forward label [↑, 3, 7, 10, 1]
Point **5**: forward label [↑, 2, 7, 9, 1]
Point **6**: forward label [↑, 4, 7, ∞, 1]

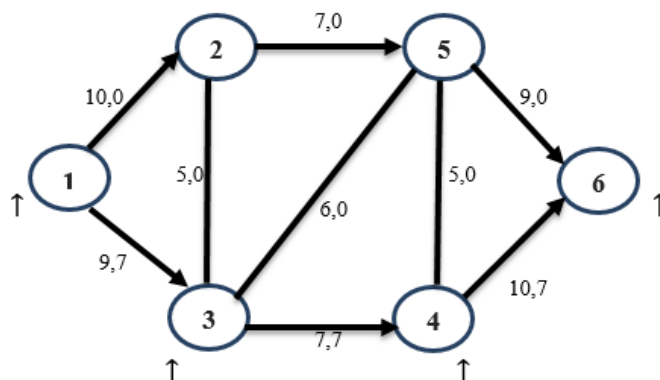Result of the flow increasing adjustment in Figure 2 and the value of the increase $v(F) = 7$.

Figure 2. The value of the increase v(F) = 7

Result of the second forward label:

Source point is **1**: forward label [↑, ∅, ∞, ∞, 1]

Point **2**: forward label [↑, 1, 10, 10, 1]

Point **3**: forward label [↑, 1, 2, 2, 1]

Point **4**:

Point **5**: forward label [↑, 2, 7, 9, 1]

Point **6**: forward label [↑, 5, 7, ∞, 1]

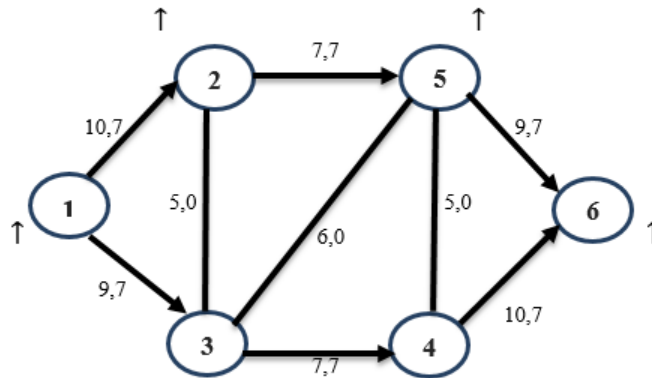Result of the flow increasing adjustment in Figure 3 and the value of the increase $v(F) = 14$.

Figure 3. The value of the increase $v(F) = 14$

Result of third forward label:

Source point is **1**: forward label [↑, ∅, ∞, ∞, 1]

Point **2**: forward label [↑, 1, 3, 3, 1]

Point **3**: forward label [↑, 1, 2, 2, 1], [↑, 2, 2, 2, 1]

Point **4**:

Point **5**: forward label [↑, 3, 2, 2, 1]

Point **6**: forward label [↑, 5, 2, ∞, 1]

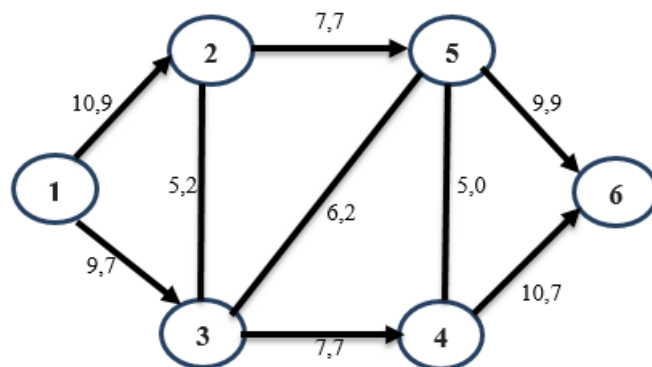Result of the flow increasing adjustment in Figure 4 and the value of the increase $v(F) = 16$.

Figure 4. The value of the increase $v(F) = 16$

This is the maximum flow, because in the following forward label is not labelled – Sink point is 6.

5.2. Example 2

Result of the first backward label:

Sink point is 6: backward label [↓, ϕ, ∞, ∞, 1]

Point 5: backward label [↓, 6, 9, 9, 1]

Point 4: backward label [↓, 6, 10, 10, 1]

Point 3: backward label [↓, 4, 7, 9, 1]

Point 2: backward label [↓, 5, 7, 10, 1]

Point 1: backward label [↓, 3, 7, ∞, 1]

Result of the flow increasing adjustment and the value of the increase $v(F) = 7$

Result of the second backward label:

Sink point is **6**: backward label [↓, ϕ, ∞, ∞, 1]

Point 5: backward label [↓, 6, 9, 9, 1]

Point 4: backward label [↓, 5, 5, 3, 1]

Point 3: backward label [↓, 5, 6, 2, 1]

Point 2: backward label [↓, 5, 7, 10, 1]

Point 1: backward label [↓, 2, 7, ∞, 1]

Result of the flow increasing adjustment and the value of the increase $v(F) = 14$

Result of third backward label:

Sink point is 6: backward label [↓, ϕ, ∞, ∞, 1]

Point 5: backward label [↓, 6, 2, 2, 1]

Point 4: backward label [↓, 6, 3, 3, 1]

Point 3: backward label [↓, 5, 2, 2, 1]

Point 2: backward label [↓, 3, 2, 3, 1]

Point 1: backward label [↓, 2, 2, ∞, 1]

Result of the flow increasing adjustment and the value of the increase $v(F) = 16$

This is the maximum flow, because in the following backward label is not labelled-Source point is 1.

6. CONCLUSION

The article regarding building a model of an extended network so that the stylization of practical problems can be applied more accurately and effectively. Next, source toward sink and sink toward source algorithm finding maximal flows are being built. Finally, a concrete example is presented to illustrate source toward sink and sink toward source algorithm.

REFERENCES

[1]   Tran Quoc Chien, "Weighted Source-Sink Alternative Algorithm to Find Maximal Flow," *The University of Danang-Vietnam Journal of Science and Technology*, vol. 3, no. 26, pp. 99-105, 2008.

[2]   Tran Ngoc Viet; Tran Quoc Chien; and Le Manh Thanh, "The Revised Ford-Fulkerson Algorithm Finding Maximal Flows on Extended Networks," *International Journal of Computer Technology and Applications*, vol. 5, no. 4, pp. 1438-1442, 2014.

[3]   Viet Tran Ngoc; Chien Tran Quoc; and Tau Nguyen Van, "Improving Computing Performance for Algorithm Finding Maximal Flows on Extended Mixed Networks*," Journal of Information and Computing Science*, England, UK, vol. 10, no. 1, pp. 075-080, 2015.

[4]   Goldberg A. V; and Tarjan R. E, "A new approach to the maximum flow problem," *Journal of the ACM (JACM)*, New York, NY, USA, vol. 35, no. 4, pp. 921-940, 1988.

[5]   Tran Quoc Chien; Nguyen Mau Tue; and Tran Ngoc Viet, "Shortest Path Algorithm on Extended Graphs," *Proceeding of the 6th National Conference on Fundamental and Applied Information Technology Research (FAIR)*, Vietnam, pp. 522-527, 2013.

[6]   Ellis L. Johnson, George L. Nemhauser; Joel S. Sokol, and Pamela H. Vance, "Shortest Paths and Multicommodity Network Flows," A Thesis Presented to the Academic Faculty, pp. 41-73, 2003.

[7]   Anderson R. J; and Jo A. C. S, "On the Parallel Implementation of Goldberg's Maximum Flow Algorithm," *Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures, New York, USA: ACM*, pp. 168-177, 1992.

[8]   Xiaolong Ma and Jie Zhou, "An extended shorted path problem with switch cost between arcs," *Proceedings of the international multiconference of engineers and computer scientists*, Hong Kong, IMECS, 2008.

[9]   Xiangming Yao, Baomin Han, Baomin Han, Hui Ren, "Simulation-Based Dynamic Passenger Flow Assignment Modelling for a Schedule-Based Transit Network;" *Discrete Dynamics in Nature and Society*- Hindawi, 2017.

[10]   Michiel C.J. Bliemer, Mark P.H. Raadsen, "Static traffic assignment with residual queues and spillback, University of Sydney," *ETH Zurich Conference paper STRC*, 2017.

[11]   P. T. Sokkalingam, R. K. Ahuja and J. B. Orlin, "New Polynomial-Time Cycle- Canceling Algorithms For Minimum-Cost Flows," *Networks*, pp. 1-21, 1996.

[12]   L. K. Fleischer, "Approximating fractional multicommodity flow independent of the number of commodities," *SIAM J. Discrete Math*., vol.13, no.4, 2000.

[13]   G. Karakostas, "Faster approximation schemes for fractional multicommodity flow problems," In Proceedings, *ACM-SIAM Symposium on Discrete Algorithms*, vol.4, no.1, 2002.

[14]   Joseph Prashker and Shlomo Bekhor, "Some observations on stochastic user equilibrium and system optimum of traffic assignment," *Transportation Research Part B: Methodological*, vol. 34, no. 4, 2000.

[15]   Luca Trevisan, "Generalizations of the Maximum Flow Problem," *Stanford University-CS261: Optimization*, 2011.

[16]   Wardrop, J. G, "Some Theoretical Aspects of Road Traffic Research," *Proceedings of the Institute of Civil Engineers*, Part II, 1952.

[17]   Aleksander, "Faster Approximation Schemes for Fractional Multicommodity Flow Problems via Dynamic Graph Algorithms," *Massachusetts Institute of Technology*, 2009.

[18]   Juan Carlos Munoz, Jorge A. Laval, "System optimum dynamic traffic assignment graphical solution method for a congested freeway and one destination," *Transportation Research*, Part B 40, 2006.

[19]   Olaf Jahn, Rolf H. Möhring, "System Optimal Routing of Traffic Flows with User Constraints in Networks with Congestion," Vol. 53, No. 4, 2005.

[20]   Erera, A.L., Daganzo, C.F., Lovell, "The access control problem on capacitated FIFO networks with unique O-D paths is hard," *Operations Research*, vol. 50, no. 4, 2002.

[21]   Laval, J.A., Mun˜oz, J.C, "System Optimum Diversion of Congested Freeway Traffic," *Institute of Transportation Studies Research Report UCB-ITS-RR*, 2002.

[22]   Ziliaskopoulos, A.K, "A linear rogramming model for the single destination system optimum dynamic traffic assignment problem," *Transportation Science*, vol. 34, no.1, 2000.

[23]   Mehlhorn, K., M. Ziegelmann, "Resource constrained shortest paths," M. Paterson, ed. Proc. 8th Annual Eur. Sympos. on Algorithms (ESA). *Lecture Notes in Computer Science*, Springer, Heidelberg, Germany, 2000.

[24]   Merchant, D. K., G. L. Nemhauser, "A model and an algorithm for the dynamic traffic assignment problems," *Transportation Sci*. 12, 1978.

[25]   Nagarney A, "Mathematical Models of Transportation and Networks," *Mathematical Models in Economics*, 2007.

[26]   Schulz, A. S., N. E. Stier-Moses, "On the performance of user equilibria in traffic networks," Proc. 14th *Annual ACM-SIAM Sympos on Discrete Algorithms (SODA). SIAM*, Philadelphia, PA, 2003.

BIOGRAPHIES OF AUTHORS

Dr. Tran Ngoc Viet. Born in 1973 in Thanh Khe District, Da Nang City, Vietnam. He graduated from Maths_IT faculty of Da Nang university of science. He got master of science at university of Danang and hold Ph.D Degree in 2017 at Danang university of technology. His main major: Applicable mathematics in transport, parallel and distributed process, discrete mathematics, graph theory, grid Computing and distributed programming

Le Hong Dung was born in 1977. He graduated from information technology faculty of Da Lat university. He got master of Danang university. Since 2018, he has been a PhD candidate at university of Danang, Vietnam. His main major: discrete mathematics, graph theory and distributed programming.