ALGORITHM FINDING MAXIMUM CONCURRENT MULTICOMMODITY LINEAR FLOW WITH LIMITED COST IN EXTENDED TRAFFIC NETWORK WITH SINGLE REGULATING COEFFICIENT ON TWO-SIDE LINES

Tran Quoc Chien and Nguyen Dinh Lau

The University of Education, University of Danang, Danang, Vietnam

ABSTRACT

Graphs and extended networks are is powerful mathematical tools applied in many fields as transportation, communication, informatics, economy, … Algorithms to find Maximum Concurrent Multicommodity Flow with Limited Cost on extended traffic networks are introduced in the works we did. However, with those algorithms, capacities of two-sided lines are shared fully for two directions. This work studies the more general and practical case, where flows are limited to use two-sided lines with a single parameter called regulating coefficient. The algorithm is presented in the programming language Java. The algorithm is coded in programming language Java with extended network database in database management system MySQL and offers exact results.

KEYWORDS

Graph, Network, Multicommodity Flow, Optimization, Approximation.

1. INTRODUCTION

Graphs are useful mathematical tools applied in various fields such as transportation, communication, information technology, economics and the others. So far in the graph there has been great and separate concern of the weight of the edges and vertices, in which the length of the road is simply the total of the weight of the edges and vertices. However, in many practical problems, weights at a vertice are different with various routes and depend on arrival edges and departure edge. For example, the period of time to go through the intersection of network traffic depends on the direction of movement of vehicles: turn right, go straight or turn left, even go to restricted direction.

The works [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25] built up maximum flow problem in the network without the capacity of vertex and cost at vertex. The above mentioned works don't mention single regulating coefficient on two-side lines either.

Our article is far optimal than the above works because of two main reasons. First, we add the capacity of vertex in network to solve the problems in real situation, section 4 is devoted **to** illustrate the examples and our solutions. Secondly, we use single regulating coefficient on two-side lines (Y) to solve two direction traffic network problems in reality.

The work [4] constructs extended traffic network model and maximum concurrent multicommodity linear flow problem with limited cost to build model for the actual problems to have accurate and efficient solutions. Algorithms to find approximate maximum concurrent multicommodity linear flow with limited cost are developed in the works [4, 5]. Nevertheless, in this algorithm, the capacities of the routes are shared fully for both directions. In some cases, vehicle of one direction will occupy a two-way route, causing traffic digestion. In this work, capacity of traffic flow on the reserve direction route will be regulated by parameter Y. Y is so-called the regulating coefficient in two-way route. Regulating coefficient Y ranges from 0.5 to 1. If Y = 0.5, the vehicles must not encroach upon the other side of the street. This is consistent with the principles of solid lines. If Y> 0.5, vehicles from reverse direction can encroach upon the other side of the street with fixed rate regulated by regulating coefficient Y.

The main result of the article is the approximation algorithm to find maximum concurrent multicommodity linear flow with limited cost on two-side lines. The algorithm is coded in programming language Java with extended network database in database management system MySQL and offers exact results.

2. MAXIMUM CONCURRENT MULTICOMMODITY LINEAR FLOW WITH LIMITED COST PROBLEM

Given a mixed graph G = (V, E) with node set V and edge set E. The edges can be directed or undirected. There are many kinds of vehicles circulating in the graph. The undirected edges are assigned two-way route and commodities with the same line in the opposite direction shares the capacities. Let us consider the following functions [1], [6], [7], [8], [9], [10].

Edge capacity function c_E:$E \rightarrow R^*$, $c_E(e)$ is the capacity of edge $e \in E$.
Node capacity function c_V:$V \rightarrow R^*$, $c_V(u)$ is the capacity of node $u \in V$.

b_E cost function: $E \rightarrow R^*$, $b_E(e)$ is the cost we have to pay to transfer a commodity to edge e.
Note that on two-side lines, the cost for the two directions might be different. $v \in V$, labeled as E_v, is the edge set belonging to node v.

b_V cost function:$V \times E_v \times E_v \rightarrow R^*$, $b_V(u,e, e')$ là the cost we have to pay to transfer a commodity from line e via link node u to line e'.

Set (V, E, c_E, c_V, b_E, b_V) is called extended traffic network.

Given that p is the path from node u via link edges e_i to node v, i = 1…(h+1), and nodes u_i, i = 1…h,

$p = [u, e_1, u_1, e_2, u_2, \ldots, e_h, u_h, e_{h+1}, v]$

The cost of circulating of a commodity on the path p is defined as:

$$b(p) = \sum_{i=1}^{h+1} b_E(e_i) + \sum_{i=1}^{h} b_V(u_i, e_i, e_{i+1}) \qquad \textbf{(1)}$$

Given an extended traffic graph G = (V, E, c_E, c_V, b_E, b_V). It is assumed that G has k pairs of source-sink nodes (s_j, t_j), Each pair is assigned with commodity j, j=1..k. Π_j is a path set from s_j to t_j in G, j = 1..k, and for $\Pi = \bigcup_{j=1}^{k} \Pi_j$.

For each path $p \in \Pi_j$, j=1..k, x(p) is commodity j circulating on path p.
Π_e refers to a set of paths in Π visting edge e, $\forall e \in E$.
Π_v refers to a set of paths in Π visiting node v, $\forall v \in V$.

$F = \{x(p) \mid p \in \Pi_j, j=1..k\}$
is defined as multicommodity flow in extended traffic network if it it satisfies the following capacity constraint:

$$\sum_{p \in \Pi_e} x(p) \le c_E(e), \forall e \in E \tag{2}$$

$$\sum_{p \in \Pi_v} x(p) \le c_V(v), \forall v \in V \tag{3}$$

Expression

$$v_j = \sum_{p \in \Pi_j} x(p), j = 1..k \tag{4}$$

is defined as the value of flow F for source-sink pair (s_j,t_j).

Given an extended traffic graph G = (V, E, c_E, c_V, b_E, b_V). It is assumed that G has k pairs of source-sink nodes (s_j, t_j), Each pair is assigned with commodity j, j=1, ….., k. j single commodity requests flow value d(j) from source node s_j to sink node t_j, $\forall j$ = 1, ..., k. Let B denote as the limited cost. The goal of the problem is to find the maximum value λ so that there is an available maximum multicommodity to transfer flow value $\lambda.d(j)$ with commodity j through flow, $\forall j$ = 1, ..., k. Simultaneously, the total cost of flow does not exceed limited cost B.

The problem is illustrated in linear program mode (P) as following:

$$\left.\begin{array}{l} \lambda \rightarrow \quad \max \\ \text{Then:} \\ \sum_{p \in \Pi_e} x(p) \le c_E(e), \forall e \in E \\ \sum_{p \in \Pi_v} x(p) \le c_V(v), \forall v \in V \\ \sum_{p \in \Pi_j} x(p) \ge \lambda.d(j), \forall j = 1..k \quad \sum_{p \in \Pi} b(p).x(p) \le B \\ x \ge 0, \lambda \ge 0 \end{array}\right\} \quad (P)$$

3. ALGORITHM TO FIND MAXIMUM CONCURRENT MULTICOMMODITY LINEAR FLOW WITH LIMITED COST ON EXTENDED TRAFFIC NETWORK WITH SINGLE REGULATING COEFFICIENT ON TWO-SIDE LINES

◊ Input:
1. Extended network $G = (V, E, c_E, c_V, b_E, b_V)$, $n=|V|$, $m=|E|$.
2. Demand (s_j, t_j, d_j), $j=1..k$.
3. Limited cost B.
4. Approximation coefficient $\omega > 0$.
5. Regulating coefficient $Y \in [0.5; 1]$.

◊ Output:
1. Maximum λ : λ_{max}
2. Actual flow:
$\{fe_j(a), fv_j(u,e,e')| a\in E, (e,u,e')\in$ Bảng b_v, $j=1,...,k\}$.
3. Actual cost $B_f \le B$.

◊ Algorithm:
// Initialize values

Let $\varepsilon = 1 - \sqrt[3]{\dfrac{1}{1+\omega}}$; $\delta = \left(\dfrac{m+n+1}{1-\varepsilon}\right)^{-\frac{1}{\varepsilon}}$;

$le(e) = \delta / c_E(e)$, $\forall e \in E$; $lv(v) = \delta / c_V(v)$, $\forall v \in V$;
$\varphi = \delta / B$;
$D = (m+n+1)\delta$;
$fe_j(a) = 0$; $\forall a\in E$,
$fv_j(u,e,e') = 0$; $\forall u\in V, \forall (e,u,e')\in b_v$, $j=1..k$
t = 1;// period count variable
$B_{ex} = 0$;// estimated cost
while D < 1 do // period
{
for j = 1 to k do // repeated cycles within j
{
d' = d_j // the amount of commodity
// flowing from s_j to t_j
while d' > 0 do // steps in each period
{

/***Algorithm finding the shortest path ([2,3]) is used to find the shortest path p from s_j to t_j and use the following length function***/

$$length(p) = \sum_{i=1}^{h+1} le(e_i) + \sum_{i=1}^{h} lv(u_i) + b(p).\varphi$$

$$=\sum_{i=1}^{h+1}\left[\varphi.b_E(e_i)+le(e_i)\right]+\sum_{i=1}^{h}\left[\varphi.b_V(u_i,e_i,e_{i+1})+lv(u_i)\right]$$

//In which

$$b(p)=\sum_{i=1}^{h+1}b_E(e_i)+\sum_{i=1}^{h}b_V(u_i,e_i,e_{i+1})$$

f' = min{d', c_E(e), c_V(v) | e∈ p, v∈ p};
B' = b(p)*f';
if B' > B {f' = f'*B/B'; B' = B};
// editing flows
fe_j(a) = fe_j(a) +f'; ∀a∈ p
fv_j(u,e,e') = fv_j(u,e,e') +f'; ∀(e,u,e')∈ p
// editing other parameters
d' = d' − f'; φ =φ*(1+ε*B'/B),
le(e) = le(e)*(1+ε*f'/c_E(e)); ∀e ∈ p
lv(v) = lv(v)*(1+ε*f'/c_V(v)); ∀v ∈ p
D = D + ε*f'*length(p);
B_{ex} = B_{ex} + B';
}//For
} //while d'
/*****Check two-way route****/
for e=(u,v) two-way
{

$$fe(u,v)=\sum_{j=1}^{k}fe_j(u,v);$$

$$fe(v,u)=\sum_{j=1}^{k}fe_j(v,u);$$

if $fe(u,v) >= Y.c_E(e).\log_{1+\varepsilon}\frac{1}{\delta}$
{
Direction u→v (assign cost b_E(u,v)=∞)
}
if $fe(v,u) >= Y.c_E(e).\log_{1+\varepsilon}\frac{1}{\delta}$
{
Direction v→u (assign cost b_E(v,u)=∞)
}
} /**** finish checking ****/

t = t + 1;
} //While D<1

// edit actual flows

$$c'=\max\left\{\frac{le(e)}{\delta/c_E(e)},\frac{lv(v)}{\delta/c_V(v)},\frac{\varphi}{\delta/B}\middle|e\in E,v\in V\right\}$$

$c_{ex} = \log_{1+\varepsilon} c'$;

$fe_j(a) = fe_j(a)/c_{ex}$; $\forall a \in E$, j=1..k
$fv_j(u,e,e') = fv_j(u,e,e')/c_{ex}$; $\forall u \in V, \forall (e,u,e') \in b_v$, j=1..k

$B_f = B_{ex} / c_{ex}$;// acutal cost

$\lambda_{\max} = \dfrac{t}{c_{ex}}$;// Maximum rate

```
// Edit reverse direction flow in two-way route
for  e ∈ E, e= (u, v)  two-way
    for  j = 1 to k do
        if  (fej(u,v)> fej(v,u)) and (fej(v,u)>0)
        {
            fej(u,v) =  fej(u,v) –  fej(v,u);
            Bf = Bf – (bE(u,v) + bE(v,u))* fej(v,u);
            fej(v,u) = 0;
        }
        if  (fej(v,u)>= fej(u,v)) and (fej(u,v)>0)
        {
            fej(v,u) =  fej(v,u) –  fej(u,v);
            Bf = Bf – (bE(u,v) + bE(v,u))* fej(u,v);
            fej(u,v) = 0;
        }
```

4. PROGRAM AND EXAMPLES

The algorithm is coded in programming language Java with extended network database in database management system MySQL.

Given an extended network as in figure 1 with 6 nodes, 6 directed edges và 3 undirected edges. The cost of edge b_E is shown in table 1 and the cost of node b_V in table 2.
The capacity of each edge is 10, the capacity for each node is là 20. There are 3 source-sink pairs (1,5), (2,4) and (3,6) within an amount of commodities.
d(1) = 10, d(2) = 10 và d(3) = 10.

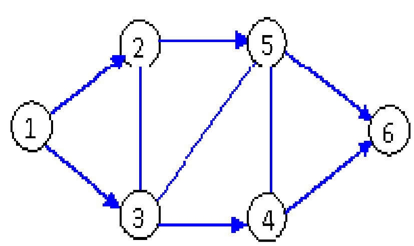

Figure 1. Extended network G

Table 1. The cost of edge b_E

Edges	b_E
(1,2)	10
(1,3)	9
(2,3)	10
(2,5)	10
(3,4)	15
(3,5)	11
(4,6)	10
(4,5)	10
(5,6)	10

Table 2. The cost of node b_V

Node	Edge 1	Edge 2	b_V
2	(1,2)	(2,3)	1
2	(1,2)	(2,5)	1
2	(3,2)	(2,5)	1
3	(1,3)	(3,4)	1
3	(1,3)	(3,5)	1
3	(1,3)	(3,2)	2
3	(5,3)	(3,2)	1
3	(5,3)	(3,4)	1
3	(2,3)	(3,4)	1
3	(2,3)	(3,5)	1
4	(3,4)	(4,6)	1
4	(3,4)	(4,5)	1
4	(5,4)	(4,6)	1
5	(2,5)	(5,3)	1
5	(2,5)	(5,4)	2
5	(2,5)	(5,6)	3
5	(3,5)	(5,4)	1
5	(3,5)	(5,6)	1
5	(4,5)	(5,3)	1
5	(4,5)	(5,6)	1

First, we consider the absence of regulatory constraints, ie regulating coefficient Y = 1.

Limited costs B = 600.
Choose approximation coefficient $\omega = 0{,}024$.
The results of maximum flow are demonstrated as following:
Maximum value $\lambda = 0{,}8968841421395084$
Actual cost: 589,6342794580955

The number of commodities is directed or distributed for each source-sink pair is 9,0**.**
Flow distribution for commodities from source 1 to sink 5, source 2 to sink 4 và source 3 to sink 6 is illustrated in figure 2, figure 3 and figure 4.

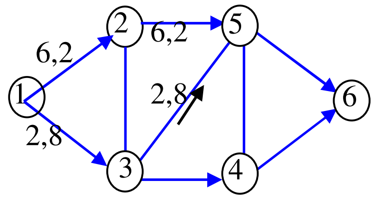

Figure 2. Flow distribution from source 1 to sink 5

It can be seen that the total of flows from node 3 to node 5 is 2,8 + 6,2 = 9,0.

Thus, the total of flows from node 3 to node 5 is nearly 10. It means that it almost encroaches upon the other side of the route from node 5 to node 3.

Now for regulating coefficient Y = 0.6.

This means that the total of flows of one direction in two-way route do not exceed 60% of the capacity.

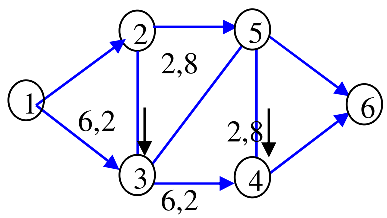

Figure 3. Flow distribution from source 2 to sink 4

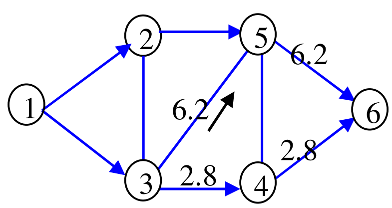

Figure 4. Flow distribution from source 3 to sink 6

Let limited cost B = 600.
Choose approximation coefficient ω = 0,024.
The results of maximum flow are demonstrated as following:

Maximum value λ = 0.798637155967107
Actual cost: 546,3976778135944

The number of commodities is directed or distributed for each source-sink pair is 8,0.

Flow distribution for commodities from source 1 to sink 5, source 2 to sink 4 and source 3 to sink 6 is illustrated in figure 5, figure 6 and figure 7.

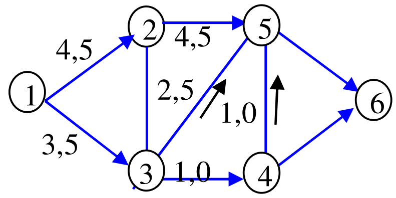

Figure 5. Flow distribution from source 1 to sink 5

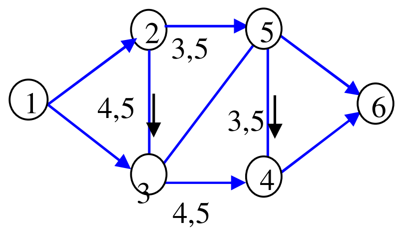

Figure 6. Flow distribution from source 2 to sink 4

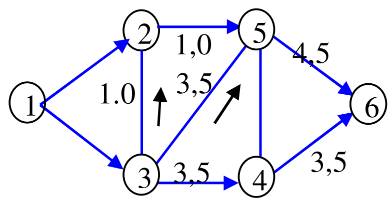

Figure 7. Flow distribution from source 3 to sink 6

It can be seen that the total of flows 8,0 is smaller than 9,0. In this case, it is impossible to regulate two-way route.

In contrast, the total of flows from node 3 to node 5 is 2,5 + 3,5 = 6,0

It is clear that the regulated flows do not exceed 60% of capacity.

5. CONCLUSION

The main result of the article is the approximation algorithm to find maximum concurrent multicommodity linear flow with limited cost on two-side lines. The algorithm is presented in the programming .

The algorithm applied to solve many practical problems can be the form of the optimization problem through extented multicommodities network, such as the traffic management problem, freight, flow distribution in the Internet, means circulation and others.

The algorithm is coded in programming language Java with extended network database in database management system MySQL and offers exact demonstrations and reliable results.

REFERENCES

[1] Tran Quoc Chien, “Extended traffic network problem and applied in Danang city”, (2011) No 507/QĐ-SKHCN date 28/12/2011

[2] Tran Quoc Chien, (2012) “Algorithms for finding the shortest paths in a general network”, journal of science and technology - university of danang, No 12(61)/2012, pp 16-21.

[3] Tran Quoc Chien, Nguyen Mau Tue, Tran Ngoc Viet, “Algorithms for finding the shortest paths in a extended network”, proceeding of the 6th national conference on fundamental and applied information technology research (FAIR), Hue, 20-21/6/2013, pp 522-527.

[4] Tran Ngoc Viet, Tran Quoc Chien, Nguyen Mau Tue, (2014) “Extended traffic network and the traffic multicommodity linear assignment problems”, journal of science and technology - university of danang, No 1(74)2014, pp 136-139.

[5] Tran Quoc Chien, (2013) “Application of the fastest path algorithm to finding maximum concurrent multicommodity linear flow with minimal cost on extended traffic network”, journal of science and technology - university of danang, No 10(71)2013, pp 85-91.

[6] Chien Tran Quoc, Lau Nguyen Dinh, Trinh Nguyen Thi Tu,(2013) “Sequential and Parallel Algorithm by Postflow-Pull Methods to Find Maximum Flow”, Proceedings 2013 13th International Conference on Computational Science and Its Applications, ISBN:978-0-7695-5045-9/13 $26.00 © 2013 IEEE, DOI 10.1109/ICCSA.2013.36, published by IEEE- CPS pp 178-181.

[7] Lau Nguyen Dinh, Chien Tran Quoc and Manh Le Thanh, (2014) “Parallel algorithm to divide optimal linear flow on extended traffic network”, Research, Development and Application on Information & Communication Technology, Ministry of Information & Communication of Vietnam, No 3, V-1, pp 14-29.

[8] Lau Nguyen Dinh, Chien Tran Quoc, Thanh Le Manh, (2014) “Improved Computing Performance for Algorithm Finding the Shortest Path in Extended Graph”, proceedings of the 2014 international conference on foundations of computer science (FCS’14), July 21-24, 2014 Las Vegas Nevada, USA, Copyright © 2014 CSREA Press, ISBN: 1-60132-270-4, Printed in the United States of America, pp 14-20.

[9] Chien Tran Quoc, Thanh Le Manh, Lau Nguyen Dinh, (2013) “Parallel algorithm to find maximum flow costlimits on extended traffic network”, Proceeding national Conference XVI "Some selected issues of Information Technology and Communications” Danang 14-15/11/2013, ISBN: 978-604-67-0251-1, pp 314-321.

[10] Lau Nguyen Dinh, Tran Quoc Chien, (2015) “Traveling Salesman Problem in Distributed Environment”, Fourth International Conference on Advanced Information Technologies and Applications (ICAITA 2015), ISSN: 2231–5403, ISBN: 978-1-921987-43-4, DOI: 10.5121/csit.2015.51501, pp 19-28.

[11] Adam N. Letchford, Juan-Jose Salazar-Gonzalez, (2014) “Stronger Multi-Commodity Flow Formulations of the Capacitated Vehicle Routing Problem”, Department of Management Science, Lancaster University, Lancaster LA1 4YW, U.K, 06/20/2014.

[12] Naveen Garg, Jochen Könemann: (2007) “Faster and Simpler Algorithms for Multicommodity Flow and Other Fractional Packing Problems”, SIAM J. Comput, Canada, 37(2), pp. 630-652.

[13] G. Karakostas, (2002) “Faster approximation schemes for fractional multicommodity flow problems”, In Proceedings, ACM-SIAM Symposium on Discrete Algorithms, pp 166–173.

[14] Marc Pfetsch, Zuse Institute Berlin, (2006) “Multicommodity Flows and Column Generation”, Technische Universität Berlin Fakultät II, Institut für Mathematik WS 2006/07.

[15] Clifford Stein, (1992) “Approximation algorithms for multicommodity flow and shop scheduling problems”, aboratory for computer science.

[16] T. Radzik. (1995) “Fast deterministic approximation for the multicommodity flow problem”, In Proceedings, ACM-SIAM Symposium on Discrete Algorithms, pp 486–492.

[17] F. Shahrokhi and D. Matula, (1990) “The maximum concurrent flow problem”, J. ACM, 37(2), pp 318–334.

[18] C. Stein, (1992) “Approximation algorithms for multicommodity flow and scheduling problems”, PhD thesis, MIT.

[19] Tamás Király and Júlia Pap, (2013) “Stable Multicommodity Flows”, journal MDPI, ISSN 1999-4893, pp 161-168.

[20] Guy Even, Moti Medina, (2012) “Online Multi-Commodity Flow with High Demands”, arXiv:1201.5030v4 [cs.DS].

[21] Maxim A. Babenko, Alexander V. Karzanov, (2012) “On Weighted Multicommodity Flows in Directed Networks”, arXiv:1212.0224v1 [math.CO].

[22] Stephan Winter, (2002) “Route Specifications with a Linear Dual Graph”, Institute for Geoinformation, Vienna University of Technology, Gusshausstr. 27-29, 1040 Vienna, Austria.

[23] Delling, D. Sanders, P.Schultes, D.Wagner, (2009) “Engineering Route Planning Algorithm”. Volume 5515 of Lecture Notes in Computer Science. Springer, pp 117–139.

[24] Volker. L, (2008) “Route Planning in Road Networks with Turn Costs” Student Research Project. (http://algo2.iti.unikarlsruhe.de/documents/routeplanning/volker_sa.pdf)

[25] Robert Geisberger and Christian Vetter, (2011) “Efficient Routing in Road Networks with Turn Costs”, Karlsruhe Institute of Technology, 76128 Karlsruhe, Germany.

Authors

Ass. Prof. DrSc. Tran Quoc Chien (http://scv.ued.udn.vn/ly_lich/chi_tiet/275).He has 14 papers in SCIE of Journal (http://www.kybernetika.cz/contact.html). Born in 1953 in Dien Ban, Quang Nam, Vietnam. He graduated from Maths_IT faculty. He got Ph.D Degree of maths in 1985 in Charles university of Prague, Czech Republic and hold Doctor of Science in Charles university of Prague, Czech Republic in 1991. He received the tittle of Ass. Pro in 1992. He work for university of Danang, Vietnam. His main major: Maths and computing, applicable mathematics in transport, maximum flow, parallel and distributed process, discrete mathemetics, graph theory, grid Computing, distributed programming.

Dr. NGUYEN DINH LAU (http://scv.udn.vn/ndlau).Born in 1978 in Dien Ban, Quang Nam, Vietnam. He graduated from Maths_IT faculty of Hue university of science in 2000. He got master of science (IT) at Danang university of technology and hold Ph.D Degree in 2015 at Danang university of technology. His main major: Applicable mathematics in transport, parallel and distributed process, discrete mathemetics, graph theory, grid Computing and distributed programming.