THE MINISTRY OF EDUCATION AND TRAINING

THE UNIVERSITY OF DANANG

TRAN NGOC VIET

THE TRAFFIC ASSIGNMENT PROBLEM AND APPLICATIONS

Major : COMPUTER SCIENCE

Code : 62 48 01 01

DOCTORAL DISSERTATION

(EXECUTIVE SUMMARY)

Danang 2017

The doctoral dissertation has been finished at:

THE UNIVERSITY OF DANANG

Advisors: 1) Assoc Prof Dr. Tran Quoc Chien

2) Assoc Prof Dr. Le Manh Thanh

Reviewer 1: …………………………………………………………

Reviewer 2: …………………………………………………………

Reviewer 3: …………………………………………………………

The dissertation is defended before The Assessment Committee at The University of Danang

Time: …..... h .... Date: ..../………/……….

The dissertation is available at:

- National Library of Vietnam
- Learning & Information Resources Center, The University of Danang

PREFACE

1. Significance of the study

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 mixed network that can be applied to modelling many practical problems more exactly and effectively. Therefore, necessary to build a model of the extended network so that the stylization of practical problems can be applied more accurately and effectively. Based on the results of the study of the problem regarding finding the maximum flow and extended graphs, the main contribution of this thesis is the traffic multicommodity linear assignment problems and applied.

2. Objects of the study

Investigating optimal theory, mainly focus on extended mixed networks algorithms to find the shortest and to find the maximum flow, the traffic multicommodity linear assignment problems with minimal cost.

3. Results of the study

- Suggesting a new algorithms finding maximal flows on extended traffic networks based on the actual requirements, proving soundness, the complexity of the algorithms and thesis also indicate the advantages of the new ones over previous algorithms.

- Developing experimental programs on extended traffic networks, then offering specific data to evaluate and compare the results of new algorithm finding maximal flows with traffic multicommodity linear assignment problems.

4. New findings of the study

- Building new shortest path algorithm on extended graphs. 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.

- Building new algorithm finding maximal flows on extended traffic networks, building a model of an extended mixed network is proposed so that the stylization of practical problems can be applied more accurately and effectively.

- Sink toward source algorithm finding maximal flows on extended mixed networks is being built and a concrete example is presented to illustrate sink toward source algorithm.

- Building a source-sink alternative algorithm finding maximal flows on extended traffic networks. Improving computing performance for algorithm finding maximal flows on extended mixed networks is being built.

- We excute optimal multicommodity linear assignment problems on traffic network. After that, we evaluate the computation time of the algorithm of the traffic multicommodity linear assignment problems.

5. Table of contents

Besides preface, conclusion, references, the thesis has three chapters:

Chapter 1: Graph, network, flow

Chapter 2: Problem finding maximal flows on extended traffic networks

Chapter 3: The traffic multicommodity linear assignment problems and applications

CHAPTER 1. GRAPH, NETWORK AND FLOW

1.1 Graph

1.2 Network

1.3 Flow

Given an network $G = (V, E, c)$, $c(i, j) \geq 0, (i, j) \in E$.

Set: $\{f(i, j) \mid (i, j) \in E\}$

is called the flow of network G if the requirements are met:

(i) $0 \leq f(i, j) \leq c(i, j), \ \forall (i, j) \in E$

(ii) Any value of point k is referring to neither a sourse point nor a sink point

$$\sum_{(i,k)\in E} f(i,k) = \sum_{(k,j)\in E} f(k, j)$$

1.4 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:

(i) $0 \leq f(x, y) \leq c_E(x, y) \quad \forall (x, y) \in E$

(ii) Any value of point z is referring to neither a sourse point nor a sink point

$$\sum_{(v,z)\in E} f(v,z) = \sum_{(z,v)\in E} f(z,v)$$

(iii) Any value of point z is referring to neither a sourse point nor a sink point

$$\sum_{(v,z)\in E} f(v,z) \leq c_V(z)$$

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

CHAPTER 2. PROBLEM FINDING MAXIMAL FLOWS ON EXTENDED TRAFFIC NETWORKS

2.1 Shortest path algorithm on extended graphs

2.1.1 Shortest path problem

2.1.2 Algorithm

+ Input: Extended graphs $G = (V, E, w_E, w_V)$, $s \in V$, $U \subset V$.

+ Output: l(v) is the length of the shortest path from s to v (if $l(v) < +\infty$), $\forall v \in U$.

+ Steps.

2.1.3 For example

2.2 Algorithm finding maximal flows on extended traffic networks

2.2.1 Introduction

2.2.2 Algorithm

+ Input: Given an extended mixed 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.

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

+ Steps:

(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 4 components.

$L_1(v)$ = [$prev_1(v)$, $c_1(v)$, $d_1(v)$, $bit_1(v)$] and can be label for the second time

$L_2(v)$ = [$prev_2(v)$, $c_2(v)$, $d_2(v)$, $bit_2(v)$].

Put labeling: $L_1(s) = [\phi, \infty, \infty, 1]$

Begin: $S = \{s\}$, $S' = \phi$

(2) Label generate

(2.1) Choose label point:

- Case $S != \phi$: Choose the point $u \in S$ of a minimum value. Remove the u from the set S, $S = S \setminus \{u\}$. Assuming that the backward label of v is [$prev_i(v)$, $c_i(t)$, $d_i(t)$, $bit_i(t)$], i = 1 or 2. A is the set of the points which are not backward label time and adjacent to the backward label point u. Step (2.2).

- Case $S == \phi,\ S' != \phi$: Assign $S = S',\ S' = \phi$. Return to step (2.1).

- Case $S == \phi,\ S' == \phi$: The flow F is the maximum. End.

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

- Case $A == \phi$: Return to step (2.1).

- Case $A != \phi$: Choose $v \in A$ of a minimum value.

Remove the v from the set A, $A = A \setminus \{v\}$.

(i) (u, v) the road section whose direction runs from u to v

If $bit_i(u) == 1$ and $f(u,v) < c_E(u,v)$ put 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 $d_i(u) > 0$;

$d_j(v) = c_V(v) - \sum\limits_{(i,v)\in E} f(i,v)$;

$bit_j(v) = 1$, if $d_j(v) > 0$,

$bit_j(v) = 0$, if $d_j(v) == 0$.

(ii) (v, u) the road section whose direction runs from v to u

If $f(v,u) > 0$, put label point v:

$prev_j(v) = u$; $c_j(v) = \min\{c_i(u), f(v,u)\}$,

$d_j(v) = c_V(v) - \sum\limits_{(i,v)\in E} f(i,v)$; $bit_j(v) = 1$.

(iii) (u, v) non-direction roads:

If $f(v,u) > 0$, put label point v case (ii).

If $f(v,u) == 0$ and $f(u,v) \geq 0$, put label point v case (i).

(3) Making adjustments in increase of the flow

(3.1) Start

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

(3.2) 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 $f(y,x) = f(y,x) - \delta$.

(3.3) Moving backwards

(i) Case x == s: Remove all the labels of the network points, except for the source point s, and return to step (2).

(ii) Case $x \neq s$: Put $y = x$

If $x = prev_2(y)$ and remove the label $L_2(y)$;

If y does not have the label $L_2(y)$ then put $x = prev_1(y)$.

Return to step (3.2).

2.2.3 For example

2.3 Source-sink alternative algorithm finding maximal flows on extended traffic networks

2.3.1 Algorithm

+ Input: Given an extended mixed network $G = (V, E, c_E, c_V)$, a source point a and a sink point z. The points in graph G are arranged in a certain order.

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

(1) Start:

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

Points from the source points and sink points will gradually be labelled L_1 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 label $(\uparrow)$ for the second time

$L_2(v) = [\uparrow, prev_2(v), c_2(v), d_2(v), bit_2(v)]$,

Form backward label:

$L_1(v) = [\downarrow, prev_1(v), c_1(v), d_1(v), bit_1(v)]$ and can be label $(\downarrow)$ for the second time

$L_2(v) = [\downarrow, prev_2(v), c_2(v), d_2(v), bit_2(v)]$,

Put labeling $(\uparrow)$ for source point and labeling $(\downarrow)$ for sink point:

$$a[\uparrow, \phi, \infty, \infty, 1] \;\&\; z[\downarrow, \phi, \infty, \infty, 1]$$

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 $S := \{a\}, S' := \phi$

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 $T := \{z\}, T' := \phi$

(2) Forward label generate:

(2.1) Choose forward label point:

• Case $S \neq \varnothing$: 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 $[\uparrow, prev_i(u), c_i(v), d_i(v), bit_i(v)]$, $i = 1$ 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 $S = \varnothing$ and $S' \neq \varnothing$: Assign $S := S'$, $S' := \varnothing$. Step (3).

• Case $S = \varnothing$ and $S' = \varnothing$: 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 u

• Case $A = \varnothing$: Return to step (2.1).

• Case $A \neq \varnothing$: Choose $t \in A$ of a minimum value. Remove the t from the set A,

$A := A \setminus \{ t \}$. Assign forward labeled point t:

If $(u,t) \in E$, $f(u,t) < c_E(u,t), bit_i(u) = 1$ put forward labeled point t:

$prev_j(t) := u$;

$c_j(t) := \min\{c_i(u),\ c_E(u,t) - f(u,t)\}$, if $d_i(u)=0$,

$c_j(t) := \min\{c_i(u),\ c_E(u,t) - f(u,t), d_i(u)\}$,

if $d_i(u) > 0$; $d_j(t) := c_V(t) - \sum\limits_{(i,t)\in G} f(i,t)$;

$bit_j(t) := 1$, if $d_j(t) > 0$,

$bit_j(t) := 0$, if $d_j(t) = 0$.

If $(t,u) \in E$, $f(t,u) > 0$ put forward labeled point t:

$prev_j(t) := u$;

$c_j(t) := \min\{c_i(u),\ f(t,u)\}$,

$d_j(t) := c_V(t) - \sum\limits_{(i,t)\in G} f(i,t)$; $bit_j(t) := 1$.

If t is not forward label, then return to step (2.2).

If t is forward label and t is backward label, then making adjustments in increase of the flow. Step (4).

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

(3) Backward label generate

(3.1) Choose backward label point:

• Case $T \neq \varnothing$: 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 (3.2).

• Case $T = \varnothing$ and $T' \neq \varnothing$: Assign $T := T'$, $T' := \varnothing$. Return to step (2).

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

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

• Case $B = \varnothing$: Return to step (3.1).

• Case $B \neq \varnothing$: 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: $prev_j(t) := v$;

$c_j(t) := \min\{c_i(v), c_E(t,v) - f(t,v)\}$, if $d_i(v) = 0$,

$c_j(t) := \min\{c_i(v), c_E(t,v) - f(t,v), d_i(v)\}$, if $d_i(v) > 0$;

$d_j(t) := c_V(t) - \sum_{(i,t) \in G} f(i,t)$;

$bit_j(t) := 1$, 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: $prev_j(t) := v$; $c_j(t) := \min\{c_i(v), f(v,t)\}$,

$d_j(t) := c_V(t) - \sum_{(i,t) \in G} f(i,t)$; $bit_j(t) := 1$.

If t is not backward label, then return to step (3.2).

If t is backward label and t is forward label, then making adjustments in increase of the flow. Step (4).

If t is backward label and t is not forward label, then add t to T', T' := T' ∪ {t}, and return to step (3.2).

(4) Making adjustments in increase of the flow

Suppose t is forward label [↑, $prev_i(t)$, $c_i(t)$, $d_i(t)$, $bit_i(t)$] and t is backward label

[↓, $prev_i(t)$, $c_i(t)$, $d_i(t)$, $bit_i(t)$]:

(4.1) Adjustment made from t back to a according to forward label

(4.1.1) Start

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

(4.1.2) 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 $f(y,x) := f(y,x) - \delta$.

(4.1.3) Moving

(i) Case $x = a$. Step (4.2).

(ii) Case $x \neq a$, put $y := x$ and $x:=h$, h is the second component of the forward labeled point y. Then return to step (4.1.2).

(4.2) Adjustment made from t back to z according to backward label

(4.2.1) Start

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

(4.2.2) 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 $f(y, x) := f(y, x) - \delta$.

(4.2.3) Moving

(i) Case $x = z$. Step (4.3).

(ii) Case $x \neq z$, put $x := y$ and $y := k$, k is the second component of the backward labeled point x. Then return to step (4.2.2).

(4.3) Remove all the labels of the network points, except for the source point a and sink point z. Return to step (2).

2.3.2 For example

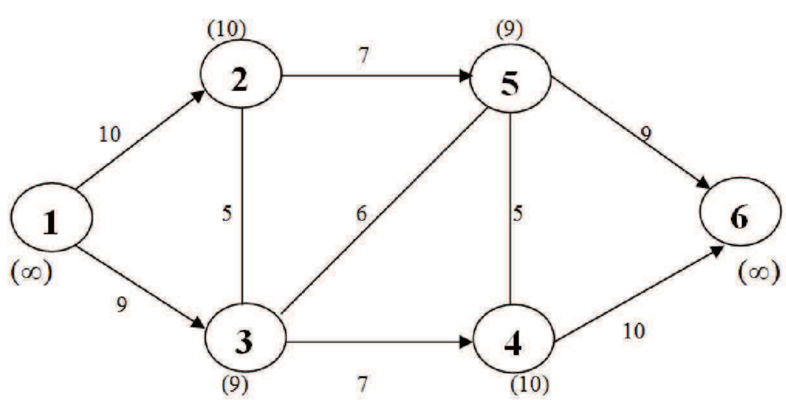

Fig. Extended mixed network

+Source point is l: [↑, ϕ, ∞, ∞, 1] and sink point is 6: [↓, ϕ, ∞, ∞, 1]

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

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

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

Point 4: forward label [↑, 3, 7, 10, 1] and backward label [↓, 6, 10, 10, 1]

+ Result of the flow increasing adjustment in figure 3 and the value of the increase v(F) = 7

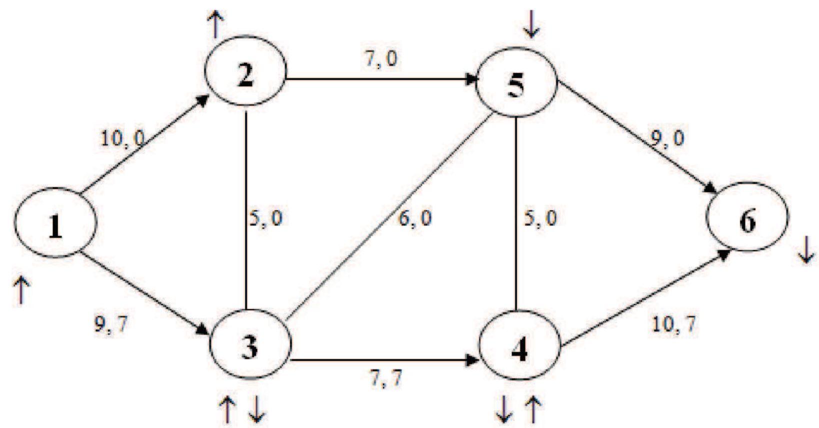

Fig. The value of the increase v(F) = 7.

+ Analog, result of the flow increasing adjustment in figure 4 and the value of the increase v(F) = 14

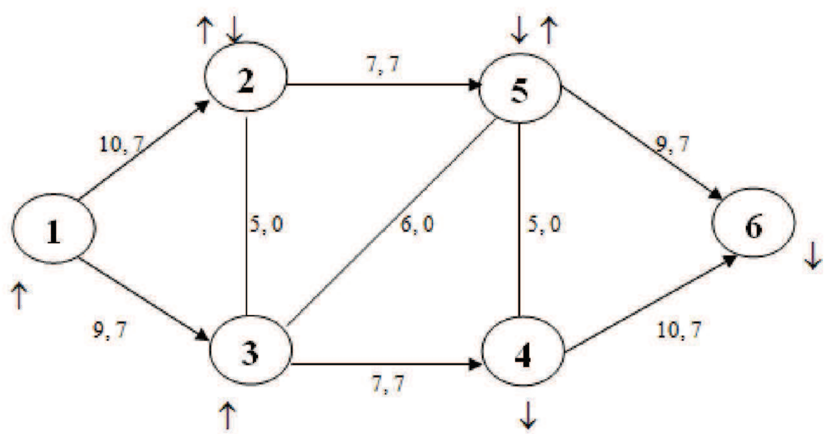

Fig. The value of the increase v(F) = 14.

+ Result of the flow increasing adjustment in figure 5 and the value of the increase v(F) = 16

This is the maximum flow, because in the following backward label and forward label is not labelled.

2.4 Sink toward source algorithm finding maximal flows on extended mixed networks

2.4.1 Algorithm

+ Input: Given an extended mixed network $G = (V, E, c_E, c_V)$, a source point a and a sink point z . The points in graph G are arranged in a certain order.

+ 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:

$L_1(v) = [\downarrow, prev_1(v), c_1(v), d_1(v), bit_1(v)]$ and can be label $(\downarrow)$ for the second time

$L_2(v) = [\downarrow, prev_2(v), c_2(v), d_2(v), bit_2(v)]$.

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

$$z[\downarrow, \phi, \infty, \infty, 1]$$

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 $T := \{z\}, T' := \phi$

(2) Backward label generate

(3) Making adjustments in increase of the flow

2.4.2 For example

Given an extended mixed network graph. 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.

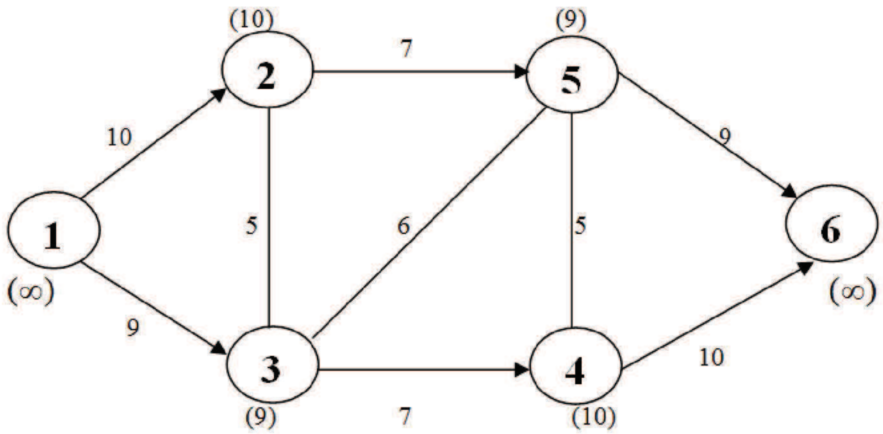

Fig. Extended mixed network

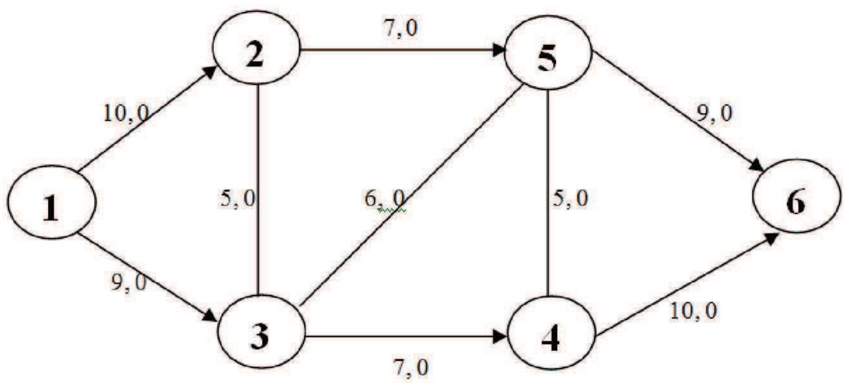

Fig. The departure flow 0

+ 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 in figure 3 and the value of the increase v(F) = 7

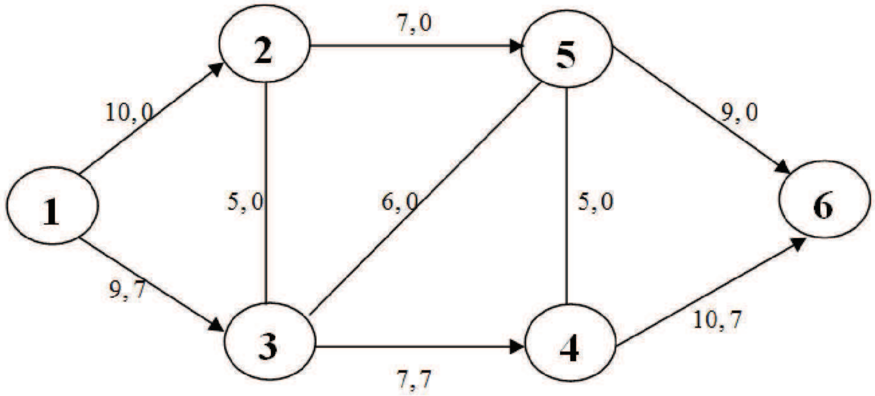

Fig. The value of the increase v(F) = 7

+ 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 in figure 4 and the value of the increase v(F) = 14

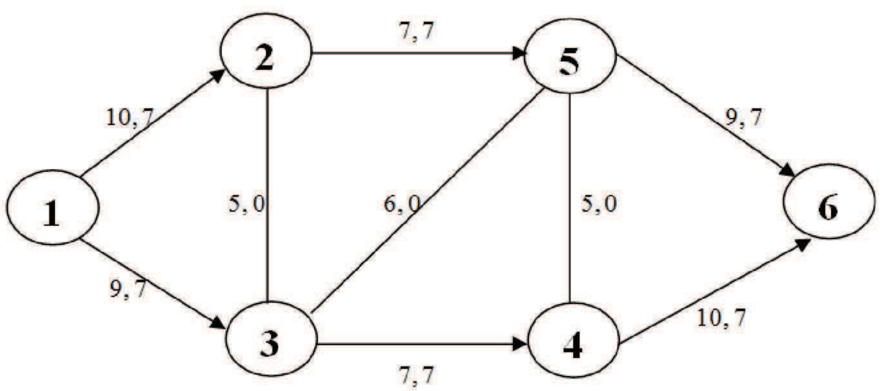

Fig. The value of the increase v(F) = 14

+ 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 in figure 5 and the value of the increase v(F) = 16

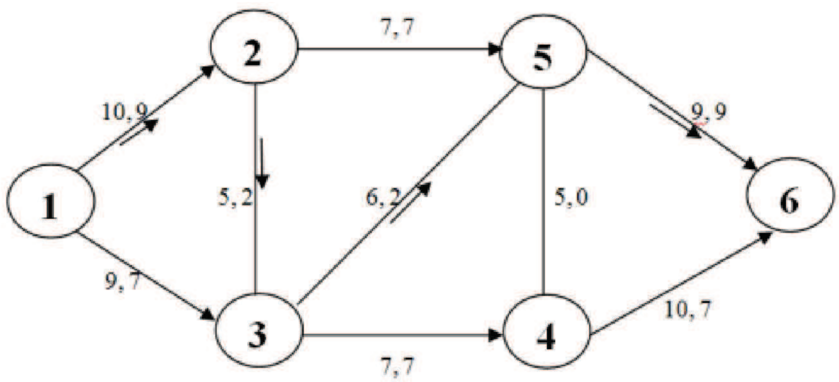

Fig. The value of the increase v(F) = 16

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

2.5 Augmenting-path maxflow algorithm on extended mixed

2.5.1 Introduction

2.5.2 Augmenting-path algorithm

2.5.3 Augmenting-path maxflow algorithm on extended mixed

2.5.4 For example

CHAPTER CONCLUSION

CHAPTER 3. THE TRAFFIC MULTICOMMODITY LINEAR ASSIGNMENT PROBLEM AND APPLICATIONS

3.1 Extended traffic networks

3.1.1 Network

A network is a mixed 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 $u \in V$.

+ The function b_E:

$E \rightarrow R^*$

+ The function b_V:

$V \times E_v \times E_v \rightarrow R^*$

+ $(V, E, c_E, c_V, b_E, b_V)$: extended traffic network.

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

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

3.1.2 The traffic multicommodity linear assignment

Extended traffic network $G = (V, E, c_E, c_V, b_E, b_V)$

$$\Pi = \bigcup_{j=1}^{k} \Pi_j$$

$$F = \{x(p) \mid p \in \Pi_j, j=1,\ldots,k\}$$

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

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

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

3.2 The traffic multicommodity linear assignment problems with minimal cost

3.2.1 Introduction

3.2.2 Algorithm

+ Input:

Extended traffic network $G = (V, E, c_E, c_V, b_E, b_V)$.

(s_j, t_j, d_j), $j=1, \dots, k$; B ; $\omega > 0$.

+ Output:

1) λ_{max}

2) $\{fe_j(a), fv_j(u,e,e') \mid a \in E, (e,u,e') \in b_V, j=1,\dots,k\}$.

3) $B_f <= B$.

+ Procedure;

Put $\varepsilon = 1 - \sqrt[3]{\dfrac{1}{1+\omega}}$;

$$\delta = \left(\frac{m+n+1}{1-\varepsilon}\right)^{-\frac{1}{\varepsilon}} ;$$

$le(e) = \delta / c_E(e),\ \forall e \in E$;

$lv(v) = \delta / c_V(v),\ \forall v \in V$;

$\varphi = \delta / B$;

$D = (m+n+1)\delta$;

$fe_j(a) = 0;\ \forall a \in E$,

$fv_j(u,e,e') = 0;\ \forall u \in V,\ \forall (e,u,e') \in b_v,\ j=1,\dots,k$

```
    t = 1;
    B_ex = 0;
while (D < 1)
    {
        for ( j = 1;  j <= k;  j ++)
          {
            d' = d_j
            while d' > 0
                {
                   length(p) = Σ_{i=1}^{h+1} le(e_i) + Σ_{i=1}^{h} lv(u_i) + b(p).φ
                   = Σ_{i=1}^{h+1} [φ.b_E(e_i) + le(e_i)] + Σ_{i=1}^{h} [φ.b_V(u_i, e_i, e_{i+1}) + lv(u_i)]
                   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
                          };
                    fe_j(a) = fe_j(a) +f'; ∀a∈p
                    fv_j(u,e,e') = fv_j(u,e,e') +f'; ∀(e,u,e')∈p
                   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';
                 }
              }
```

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

$t = t + 1;$

}

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

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

$fe_j(a) = fe_j(a)/c_{ex}; \forall a \in E, j=1,...,k$

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

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

$\lambda_{max} = \frac{t}{c_{ex}}$;

for $e \in E, e = (u, v)$

for (j = 1; j <= k; j + +)

if $(fe_j(u,v) > fe_j(v,u))$ and $(fe_j(v,u) > 0)$

{

$fe_j(u,v) = fe_j(u,v) - fe_j(v,u);$

$B_f = B_f - (b_E(u,v) + b_E(v,u)) * fe_j(v,u);$

$fe_j(v,u) = 0;$

}

if $(fe_j(v,u) >= fe_j(u,v))$ and $(fe_j(u,v) > 0)$

{

$fe_j(v,u) = fe_j(v,u) - fe_j(u,v);$

$B_f = B_f - (b_E(u,v) + b_E(v,u)) * fe_j(u,v);$

$fe_j(u,v) = 0;$

}

3.2.3 For example

3.3 The traffic multicommodity linear assignment problems and applied

3.3.1 Introduction

3.3.2 Algorithm

+ Input:

Extended traffic network $G = (V, E, c_E, c_V, b_E, b_V)$.

(s_j, t_j, d_j), $j = 1,\ldots,k$.

$\lambda_{inf} \approx 1$;

$\omega > 0$.

+ Output:

1) $\{fe_j(a), fv_j(u,e,e')| a \in E, (e,u,e') \in b_V, j=1,...,k\}$.

2) B_{min}.

+ Procedure;

```
B = 0;
 for ( j = 1;  j <= k;  j + +)
      {
         B = B + d_j*b(p);
      }
   if  (λ_max < λ_inf)
      { B = B / λ_max ;
       }
while (λ_max < λ_inf)
 if  (λ_max > 1)
       {  fe_j(a) = fe_j(a) / λ_max; ∀a∈E, j=1,...,k
          fv_j(u,e,e') = fv_j(u,e,e') / λ_max; ∀u∈V, ∀(e,u,e')∈ b_v, j=1,...,k
          B_min = B_f / λ_max;
        }
```

3.3.3 Program

3.3.4 Applied

CHAPTER CONCLUSION

CONCLUSION

Here are some main findings of the study.

Firstly, building new shortest path algorithm on extended graphs. 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.

Secondly, building new algorithm finding maximal flows on extended traffic networks, building a model of an extended mixed network is proposed so that the stylization of practical problems can be applied more accurately and effectively.

Thirdly, sink toward source algorithm finding maximal flows on extended mixed networks is being built and a concrete example is presented to illustrate sink toward source algorithm.

Next, building a source-sink alternative algorithm finding maximal flows on extended traffic networks. Improving computing performance for algorithm finding maximal flows on extended mixed networks is being built.

Finally, we excute optimal multicommodity linear assignment problems on traffic network. After that, we evaluate the computation time of the algorithm of the traffic multicommodity linear assignment problems.

FURTHER RESEARCH

- Building bigger volume of input data to have more chance to compare computation time.

- Researching in parallel algorithms finding maximal flows on extended traffic networks.

PUBLICATIONS OF THE AUTHOR

[1] Tran Ngoc Viet, Tran Quoc Chien, Le Manh Thanh, The Revised Ford-Fulkerson Algorithm Finding Maximal Flows on Extended Networks, International Journal of Computer Technology and Applications.

[2] Viet Tran Ngoc, Chien Tran Quoc, Tau Nguyen Van, Improving Computing Performance for Algorithm Finding Maximal Flows on Extended Mixed Networks, Journal of Information and Computing Science.

[3] Tran Ngoc Viet, Tran Quoc Chien, Nguyen Mau Tue, Optimal multicommodity linear assignment problems on traffic network, Proceedings of the 7th National Conference on Fundamental and Applied Information Technology Research - FAIR 2014, 31-39.

[4] Tran Ngoc Viet, Tran Quoc Chien, Le Manh Thanh, The revised Ford-Fulkerson algorithm finding maximal flows on extended networks, Proceedings of the 7th National Conference on Fundamental and Applied Information Technology Research - FAIR 2014, 643-649.

[5] Tran Quoc Chien, Nguyen Mau Tue, Tran Ngoc Viet, Shortest path algorithm on extended graphs, Proceedings of the 6th National Conference on Fundamental and Applied Information Technology Research - FAIR 2013, 522-527.

[6] Tran Quoc Chien, Tran Ngoc Viet, Nguyen Dinh Lau, Algorithm finding maximal flows on extended traffic networks, The university of Danang Journal of science and technology, RAIT 2014, 1-4.

[7] Tran Ngoc Viet, Tran Quoc Chien, Nguyen Mau Tue, Extended traffic network and the traffic multicommodity linear assignment problems, The university of Danang Journal of science and technology, 2014, 136-139.

[8] Viet Tran Ngoc, Hung Hoang Bao, Chien Tran Quoc, Thanh Le Manh, Sink Toward Source Algorithm Finding Maximal Flows on Extended Mixed Networks, ACIIDS 2016.