The Revised Ford-Fulkerson Algorithm Finding Maximal Flows On Extended Networks

Tran Ngoc Viet¹, Tran Quoc Chien¹, Le Manh Thanh²
¹The university of Danang-Vietnam, ²Hue university-Vietnam
trviet01@yahoo.com, tqchien@dce.udn.vn, lmthanh1953@yahoo.com

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 indepently 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 modelize many practical problems more exact and effective. The main contribution of this paper is the revised Ford-Fulkerson algorithm finding maximal flows on extended networks.

Key word - graph, network, flow, maximalFlow, algorithm.

I. 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 indepently 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 modelize many practical problems more exact and effective. 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 [1], [2] and extended graphs [3], the main contribution of this paper is the revised Ford-Fulkerson algorithm finding maximal flows on extended networks.

II. EXTENDED 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 \( c_E : E \rightarrow \mathbb{R}^+ \), \( c_E(e) \) the route circulation possibility \( e \in E \).

+ The function of the circle circulation possibility \( c_V : V \rightarrow \mathbb{R}^+ \), \( c_V(u) \) the circle circulation possibility \( u \in V \).
+G = (V, E, cE, cv): extended network.

III. THE FLOW OF THE EXTENDED NETWORK

Given an extended network

\[ G = (V, E, cE, cv) \]

a source point s and a sink point t.

Set: \( \{ f(x,y) \mid (x,y) \in G \} \), is called the flow of network \( G \) if the requirements are met:

(i) \( 0 \leq f(x,y) \leq cE(x,y) \) \( \forall \(x,y) \in G \)

(ii) Any value of point \( z \) is referring to

\[
\sum_{\(v,z) \in G} f(v,z) = \sum_{(z,v) \in G} f(z,v)
\]

(iii) Any value of point \( z \) is referring to

\[
\sum_{(v,z) \in G} f(v,z) \leq cv(z)
\]

Expression:

\[
v(F) = \sum_{(s,v) \in G} f(s,v),
\]

is called the value of flow \( F \).

- **The maximum problem:**

Given an extended network \( G = (V, E, cE, cv) \), 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.

- **Theorem 1:** Given an extended network \( G = (V, E, cE, cv) \), a source point s and a sink point t, then exist is the maximal flow.

Conformable proof [1].

IV. THE REVISED FORD-FULKERSON ALGORITHM FINDING MAXIMAL FLOWS ON EXTENDED NETWORKS

+ **Input:** Given an extended network

\[ G = (V, E, cE, cv), \]

a source point s and a sink point t. The points in graph \( G \) are arranged in a certain order.

+ **Out:** 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 will gradually be labelled \( L_1 \) for the first time including 4 components

\[ L_1(v) = [\text{prev}_1(v), c_1(v), d_1(v), \text{bit}_1(v)] \] and can be labelled for the second time

\[ L_2(v) = [\text{prev}_2(v), c_2(v), d_2(v), \text{bit}_2(v)] \].

Label the source point as:

\[ L_1(s) = [\varnothing, \infty, \infty, 1] \]

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

\[ S := \{ s \}, S' := \varnothing \]

(2) **Labelled:**

(2.1) Choose the labelled 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 sink of \( u \) is \([\text{prev}_i(u), c_i(v), d_i(v), \text{bit}_i(v)], i \) 1 or 2.

A is the set of the points which are not labelled for the second time and adjacent to the label point \( u \). Step (2.2).

- **Case** \( S = \varnothing \) and \( S' \neq \varnothing \): Assign \( S := S' \) and \( S' := \varnothing \). Return to step (2.1).

- **Case** \( S = \varnothing \) and \( S' = \varnothing \): The flow \( F \) is the maximum. **End.**

(2.2) Labelling the points which are not labelled and are adjacent to the label points

- **Case** \( A = \varnothing \): Return to step (2.1).

- **Case** \( A \neq \varnothing \): Choose \( v \in A \) of a minimum value. Remove the \( v \) from the set \( A \).

\[ A := A \setminus \{ v \} \]. Mark \( j \): an indicator expressing times of the labelling, \( j = 1 \) or 2 the first time or the second time accordingly.
Form a label $L_f(v)$ of the point $v$ according to each of the following cases:

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

If $bit(u)=1$ and $f(u,v) < c_E(u,v)$, label the point $v$ as follows:

$$prev_1(v) := u;$$
$$c_f(v) := \min\{c_f(u), c_E(u,v) - f(u,v)\},$$
$$d_f(v) := c_f(v) - \sum_{i \in G} f(i,v);$$
$$bit_f(v) := 1,$$ if $d_f(v) > 0;
$$bit_f(v) := 0,$$ if $d_f(v) = 0.$

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

If $f(v,u) > 0$, label the point $v$ as follows:

$$prev_2(v) := u;$$
$$c_f(v) := \min\{c_f(u), f(v,u)\},$$
$$d_f(v) := c_f(v) - \sum_{i \in G} f(i,v);$$
$$bit_f(v) := 1.$$

(iii) $(u,v)$ non-direction roads.

If $f(v,u) > 0$, label the point $v$ as in case (ii).

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

If $v$ is not labelled, then return to step (2.2).

If $v$ is labelled and $v = t$, then the step to increase the amount of the flow will be carried out (3).

If $v$ is labelled and $v \neq t$, add $v$ to $S'$, $S' := S' \cup \{v\}$, and return to step (2.2).

(3) Making adjustments in increase of the flow

If $t$ is labelled $[prev_1(t), c_1(t), d_1(t), bit_1(t)]$:

(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 $y$ is labelled $L_2(y)$, then

put $x := prev_2(y)$ và 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).

**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 $\delta$ 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.

**Theorem 3:** Given an extended network $G = (V, E, c_E, c_V)$, a source point $s$ and a sink point $t$, $F = \{f(x,y) \mid (x,y) \in G\}$ the flow will get its maximum value when the algorithm is ended. Then, $F$ is the maximum flow.
Conformable proof [1].

- 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. The number of times to increase the flow in maximum is $v^*$. So the complexity of the algorithm is $O(v^*(|E| + 2.|V|))$.

V. RESULT OF THE EXPERIMENT

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 1, the sink point is 6.

![Figure 1- extended network](image1)

![Figure 2- the departure flow 0](image2)

<table>
<thead>
<tr>
<th>Circle</th>
<th>L_1</th>
<th>L_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$[\emptyset, \infty, \infty, 1]$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$[1, 10, 10, 1]$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$[1, 9, 9, 1]$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$[3, 7, 10, 1]$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$[2, 7, 9, 1]$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>$[4, 7, \infty, 1]$</td>
<td></td>
</tr>
</tbody>
</table>

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

![Figure 3- the value of the increase $v(F) = 7.$](image3)

<table>
<thead>
<tr>
<th>Circle</th>
<th>L_1</th>
<th>L_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$[\emptyset, \infty, \infty, 1]$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$[1, 10, 10, 1]$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$[1, 2, 2, 1]$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$[2, 7, 9, 1]$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$[5, 7, \infty, 1]$</td>
<td></td>
</tr>
</tbody>
</table>

![Figure 4- the value of the increase $v(F) = 14.$](image4)

<table>
<thead>
<tr>
<th>Circle</th>
<th>L_1</th>
<th>L_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$[\emptyset, \infty, \infty, 1]$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$[1, 10, 10, 1]$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$[1, 2, 2, 1]$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$[2, 7, 9, 1]$</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Result of the first label

Table 2: Result of the second label

Table 3: Result of third label
Figure 5- the value of the increase \( \nu(F) = 16 \).

This is the maximum flow, because in the following label, the sixth point is not labelled.

VI. 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, the Ford-Fulkerson to be revised to find the maximum flow on the extended network is being built. Finally, a concrete example is presented to illustrate the revised Ford-Fulkerson.

VII. REFERENCE


