On Finding the Maximum Capacity Flow in Networks

Chong Hyung Lee[a], **Dong Ho Park and Seung Min Lee**[b]

[a]*Division of Information Technology, Konyang University, Nonsan, Korea*

[b]*Depatment of Statistics, Hallym University, Chunchon, Korea*

Abstract

An efficient method is developed to obtain the maximum capacity flow for a network when its simple paths are known. Most of the existing techniques need to convert simple paths into minimal cuts, or to determine the order of simple paths to be applied in the process to reach the correct result.

In this paper, we propose a method based on the concepts of signed simple path and signed flow defined in the text. Our method involves a fewer number of arithmetic operations at each iteration, and requires fewer iterations in the whole process than the existing methods. Our method can be easily extended to a mixed network with a slight modification. Furthermore, the correctness of our method does not depend on the order of simple paths to be applied in the process.

Key words - Flow augmenting simple path, Signed simple path, Signed flow

1. INTRODUCTION

A network is modeled as a graph $G(V, E)$, which consists of a set V of nodes and a set E of links where each link may have different capacity. To develop an efficient method for computing the maximum flow for a network with variable link-capacities has attracted a great deal of attention in the literature. Recently, a number of methods have been proposed for this purpose, especially for the evaluation of the measures closely related with network performance under the assumption that the simple paths of the network are known. The methods suggested

E-mail address: CHLee@konyang.ac.kr, DHPark@hallym.ac.kr, SMLee1@hallym.ac.kr.

by [1-2], [6] and [9] yield incorrect results for certain situations. Ref [1] mentions that the method in [2] lacks generality, and [8] also discusses the drawbacks of the methods in [1] and [9]. Ref [7] presents counter examples to show that the methods of [1] and [6] fail in certain cases. The method of [7] complements the drawbacks of the preceding results, but needs extra efforts of converting the given simple paths into minimal cuts. Ref [4] computes and keeps, at each iteration, the residual-capacity of the network and hence, the method is affected by the order in that each simple path is applied.

In this paper, we select, at each iteration, a flow augmenting simple path based on the concepts of signed simple path and signed flow defined in the text. The correctness of our method is guaranteed, regardless of the order of simple paths to be applied in the process and, for efficiency, we may select a simple path which contains the smallest number of links first. At each iteration, the selection procedure is simple and, by excluding unnecessary simple paths beforehand, the selection is made only from the set of remaining simple paths of the network. Thereby, our method involves a fewer number of arithmetic operations at each iteration, and requires fewer iterations in the whole process to compute the maximum flow of the given network than the existing methods do.

2. SIGNED SIMPLE PATH AND SIGNED FLOW

2.1 Signed Simple Path and Signed Flow

A simple path is an open edge train connecting the source node (s) and the terminal node (t), in which no node is traversed more than once. Let i be a link in the network and let P be a simple path which contains the link i. When we traverse on P from s to t node by node, the link i is uniquely expressed as an edge (a, b) or (b, a), where a and b are two incident nodes connected by link i. We say that the link i has the direction, in P, of $a \rightarrow b$ if it appears as an edge (a, b), and the direction of $b \rightarrow a$ otherwise. We call this the *link direction of* i *in* P. We note that the link direction of i in P may be the same as or opposite to that of the link i in another simple path.

When the flow of positive amount actually moves through link i, it has its moving direction. We call this the *flow direction on link* i. For an undirected network, the flow direction on link i may be either $a \rightarrow b$ or $b \rightarrow a$. We distinguish these two possible directions of $a \rightarrow b$

and $b \to a$ by the signes of '+' and '−', for example, $a \to b$ as '+' and $b \to a$ as '−', or vice versa.

Definition 1. A simple path P in which each edge is represented as a link signed by its link directon in P is said to be a *signed simple path* P. Similarly, the flow on link i which is signed by its flow direction is said to be the *signed flow on* i.

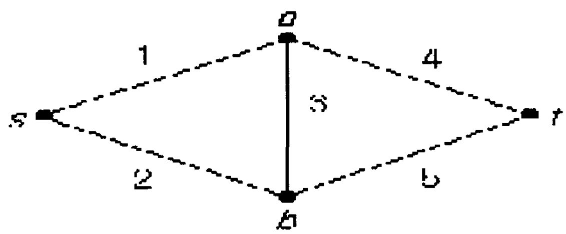

Figure 1. Bridge Network

As an example, we consider the bridge network shown in Figure 1. Defining an order on the nodes as $s < a < b < t$, we use '+' for $n \to n'$ if $n < n'$, and use '−' otherwise. The + sign may be omitted. There are four signed simple paths: (1,4), (2,5), (1,3,5), (2,−3,4). The flow of amount 10 moving $b \to a$ on link 3 is said to be the signed flow of −10 on link 3. The null flow has no sign. Hereafter, a simple path and flow mean a signed simple path and signed flow.

2.2 Flow Augmenting Simple Path

Given the capacity vector $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)$, where $c_i(> 0)$ denotes the link capacity of i, let $\boldsymbol{f} = (f_1, f_2, \cdots, f_n)$ be a feasible flow pattern of the network and let f_i be the flow on link i. If $f_i = 0$, then link i is said to be *flowless.* We consider a simple path P and a link $i \in P$ which is not flowless. The flow direction of f_i may be the same as or opposite to the link direction of i in P. We say that link i is *with forward flow* f_i in P if they are the same, and *with reverse flow* f_i otherwise. We note that each link in P is one of the following three types: with forward flow, with reverse flow, or flowless. The link i in P is said to be *saturated in* P, if f_i is forward flow on i in P and $|f_i| = c_i$.

Definition 2. A simple path P is said to be a *flow augmenting simple path* ($fasp$) with respect to $\boldsymbol{f}$ if there is no saturated link in P.

Let P be an $fasp$ with respect to $\boldsymbol{f}$. Now, let w_P be the minimum of $(c_i - |f_i|)$ taken over all links with forward flow and all flowless links in P, and let w_R be the minimum of $|f_i|$'s taken over all links with reverse flow in P. We define $w = min(w_F, w_R)$, taken only over existing terms. Then w would be the augmented amount of flow by P with respect to $\boldsymbol{f}$, and we adjust $\boldsymbol{f}$ accordingly to get a new flow pattern, $\boldsymbol{f}^*$ say. We observe that the value of net flow from s to t of a flow pattern $\boldsymbol{f}$ is the maximum flow of the network if and only if there is no more $fasp$ with respect to $\boldsymbol{f}$.

3. ALGORITHM

In this section, we present the algorithm to compute the maximum flow of a given network and exemplify the use of algorithm by solving the bridge network of Figure 1. To establish an algorithm, we start with the zero flow pattern $\boldsymbol{f} = (0, 0, \cdots, 0)$. At each iteration, we first select the simple path P with the smallest number of links, which will be referred to as the shortest simple path in the sequel. If P is an $fasp$ with respect to current $\boldsymbol{f}$, then adjust $\boldsymbol{f}$ accordingly. For each of the saturated uni-directional links in P, we remove all simple paths containing it from further consideration. The process stops when there is no more $fasp$ left with respect to the current $\boldsymbol{f}$. In algorithm, we select the shortest simple path and check if it is an $fasp$. The order of simple paths to be applied may affect the efficiency of the algorithm, but not the correctness of the result.

$\boldsymbol{c}$	link capacity vector, which is given
$\boldsymbol{f}$	current flow pattern
MF	current amount of net flow from s to t, for computing maximum capacity flow
AVSP	set of available simple paths
TEMPSP	temporary set of AVSP
$sign(x)$	integer-valued function; +1 if $x > 0$, 0 if $x = 0$, and −1 if $x < 0$.

3.1 Algorithm

1. Initialize $\boldsymbol{f} = (0, 0, \cdots, 0)$, MF= 0 and AVSP={ all simple paths };

2. TEMPSP=AVSP;

3. Select the shortest simple path P in TEMPSP;

   if P is not an $fasp$ **then**

   begin

   TEMPSP=TEMPSP$-\{P\}$;

   if TEMPSP$=\emptyset$ **then** $STOP$ **else** go to 3;

   end;

   for each $i \in P$ **do**

   if $sign(i) + sign(f_i) = 0$ **then** $w_i = |f_i|$ **else** $w_i = c_i - |f_i|$;

   Set $w = min_{i \in P} w_i$ and MF=MF+w;

   for each $i \in P$ **do**

   begin

   $f_i = f_i + sign(i) \times w$;

   if $|f_i| = c_i$ and i is uni-directional **then** AVSP=AVSP$-\{P' | i \in P'\}$;

   end;

   Go to 2;

Example 1. Consider the bridge network in Figure 1, which has four simple paths: (1,4), (2,5), (1,3,5), (2,−3,4) and all links except 3 are uni-directional. Let the capacity vector be given as $\boldsymbol{c} = (2, 6, 2, 5, 3)$. The saturated uni-directional links are marked by 's' in flow patterns. The process stops, when there is no more $fasp$ left in AVSP.

Table 1 : Process for Figure 1

$fasp$	w	capacity vector	$\boldsymbol{f}$	MF	AVSP
-	-	(2,6,2,5,3)	(0,0,0,0,0)	0	{(1,4),(2,5),(1,3,5),(2,-3,4)}
(1,4)	2		(s,0,0,2,0)	2	{(2,5),(2,−3,4)}
(2,5)	3		(s,3,0,2,s)	5	{(2,−3,4)}
(2,−3,4)	2		(s,5,−2,4,s)	7	{(2,−3,4)}

4. DISCUSSION

To evaluate the measures for network performance such as network reliability or performance index, a sequence of subnetworks are generated in succession one by one by adding certain links to the current one, and then its corresponding maximum flow is computed for each subnetwork generated. See, for example, [1], [5], [7] and [9] for references. since our method is not dependent on the order of simple paths to be applied, the earlier steps already completed for the given subnetwork need neither to be repeated nor to be altered. In consideration of the complexity of the system that many reliability engineers usually face and the great number of subnetworks generated in the evaluation process, our method would be working more efficiently than the existing methods.

REFERENCES

[1] K.K. Aggarwal, "A fast algorithm for the performance index of a telecommunication network", IEEE Trans. Reliability, vol 37, 1988 Apr, pp 65-69.

[2] K.K. Aggarwal, Y.C. Chopra, J.S. Bajwa, "Capacity consideration in reliability analysis of communication systems", IEEE Trans. Reliability, vol 31, 1982 Jun, pp 177-181.

[3] W.K. Chen, *Theory of Nets: Flows in Networks*, John Wiley & Sons, Inc., 1990.

[4] K. Kyandoghere, "A note on : reliability modeling & performance of variable link-capacity networks", IEEE Trans. Reliability, vol 47, 1998 Mar, pp 44-48.

[5] S.M. Lee, D.H. Park, "An Efficient Method for Evaluation of Network Reliability with Variable Link-Capacities", IEEE Trans. Reliability, 2001 Dec, pp 374-379.

[6] K.B. Misra, P. Prasad, "Comments on: Reliability evaluation of a flow network", IEEE Trans. Reliability, vol 31, 1982 Jun, pp 174-176.

[7] S. Rai, S. Soh, "A computer approach for reliability evaluation of telecommunication networks with heterogeneous link-capacities", IEEE Trans. Reliability, vol 40, 1991 Oct, pp 441-451.

[8] R. Schanzer, "Comment on : Reliability modeling and performance of variable link-capacity networks", IEEE Trans. Reliability, vol 44, 1995 Dec, pp 620-621.

[9] P.K. Varshney, A.R. Joshi, P.L. Chang, "Reliability modeling and performance evaluation of variable link capacity networks", IEEE Trans. Reliability, vol 43, 1994 Sep, pp 378-382.