The Korean Communications
in Statistics Vol. 9, No. 3, 2002
pp. 845-851

Finding the Maximum Flow in a Network with Simple Paths

Seung Min Lee[1], Chong Hyung Lee[2], Dong Ho Park[3]

Abstract

An efficient method is developed to obtain the maximum 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.

Keywords : Maximum flow, Flow augmenting path, Signed simple path, Signed flow

1. Introduction

Assumptions

1. The nodes are perfect and each has no capacity limit.
2. All the links are undirected and each link flow is bounded by the link capacity.
3. No flow can be transmitted through a failed link.
4. The simple paths of the network, considering connectivity only, are known.

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

---

1) Research supported by Hallym University Research Fund, 2001.
Professor, Department of Statistics, Hallym University, Chunchon 200-702, Korea
E-mail : smlee1@hallym.ac.kr

2) Full-time Instructor, Division of Information Technology, Konyang University, Nonsan 320-711, Korea

3) Professor, Department of Statistics, Hallym University, Chunchon 200-702, Korea

performance under the assumption that the simple paths of the network are known. The methods suggested by Aggarwal(1988), Aggarwal, Chopra and Bajwa(1982), Misra and Prasad(1982) and Varshney, Joshi and Chang(1994) yield incorrect results for certain situations. Aggarwal(1988) mentions that the method in Aggarwal, Chopra and Bajwa(1982) lacks generality, and Schanzer(1995) discusses the drawbacks of the methods in Aggarwal(1988) and Varshney, Joshi and Chang(1994). Rai and Soh(1991) presents counter examples to show that the methods of Aggarwal(1988) and Misra and Prasad(1982) fail in certain cases. The method of Rai and Soh(1991) complements the drawbacks of the preceding results, but needs extra efforts of converting the given simple paths into minimal cuts and computes the sum of all link capacities for each minimal cut. Kyandoghere(1998) computes and keeps, at each iteration, the residual-capacity of the network and other quantities for each simple path to determine the next simple path to be applied. 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. In Section 2, we present the necessary concepts to propose our method and a few definitions including signed simple path and signed flow. Section 3 gives the detailed descriptions on the methodology and algorithm utilizing the concepts defined in the previous section. Some examples are presented as well. In Section 4, we further discuss the advantages of the proposed method when it is used in conjunction with other processes.

2. Signed simple path and signed flow

Acronyms

ssp : signed simple path

sf : signed flow

udl : uni-directional link

fassp : flow augmenting signed simple path

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 $b \rightarrow a$ by the signs of '+' and '-', for example, $a \rightarrow b$ as '+' and $b \rightarrow a$ as '-', or vice versa.

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

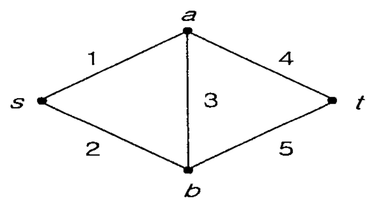

Figure 1. Bridge Network

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 $n \rightarrow n'$ if $n < n'$, and use '-' otherwise. The '+' sign may be omitted. There are four *ssp*: (1,4), (2,5), (1,3,5), (2,-3,4). The flow of amount 10 moving $b \rightarrow a$ on link 3 is said to be the *sf* of -10 on link 3. The null *sf* has no sign.

2.2 Flow Augmenting Signed 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 $f = (f_1, f_2, \cdots, f_n)$ be a feasible *sf* pattern of the network and let f_i be the *sf* on link i. If $f_i = 0$, then link i is said to be *flowless*. We consider a *ssp* 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 sf f_i* in P if they are the same, and *with reverse sf f_i* otherwise. We note that each link in P is one of the following three types: with forward *sf*, with reverse *sf*, or flowless. The link i in P is said to be *saturated in* P, if f_i is forward *sf* on i in P and $|f_i| = c_i$.

Definition 2. A *ssp* P is said to be a *fassp* with respect to f if there is no saturated link in P.

Let P be an *fassp* with respect to f. Now, let w_F be the minimum of $(c_i - |f_i|)$ taken over all links with forward *sf* and all flowless links in P, and let w_R be the minimum of $|f_i|$'s taken over all links with reverse *sf* 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 f, and we adjust f accordingly to get a new *sf* pattern, f^* say. We observe that the value of net flow from s to t of a *sf* pattern f is the maximum flow of the network if and only if there is no more *fassp* with respect to f. Such an idea and the concept of *fassp* are similar to those of *flow augmenting path* suggested for the directed networks. For detailed discussions, see Chen(1990). We note that a *ssp* P, which is not an *fassp* with respect to f, can become an *fassp* for a new *sf* pattern f^*. On certain circumstances, however, we can identify the *ssp* which would never become *fassp* and, hence, exclude them from further consideration in computing maximum flow of the network. A link is said to be a *udl*, if the link direction of it is the same in all *ssp* containing it. Suppose that a *udl* i is currently saturated in some *ssp*. Then, any *ssp* containing i would never become an *fassp*, since i remains saturated throughout. For the bridge network in Figure 1, all the links except 3 are *udl*.

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 *sf* pattern $f = (0, 0, \cdots, 0)$. At each iteration, we first select the *ssp* P with the smallest number of links, which will be referred to as the shortest *ssp* in the sequel. The selection of the shortest path reduces the number of augmentation. See, for example, Chen(1990). If P is an *fassp* with respect to current f, then adjust f accordingly. For each of the saturated *udl* in P, we remove all *ssp* containing it from further consideration. The process stops when there is no more *fassp* left with respect to the current f. In algorithm, we select the shortest *ssp* and check if it is an *fassp*. The order of *ssp* to be applied may affect the efficiency of the algorithm, but not the correctness of the result.

Notation for Algorithm

c	link capacity vector, which is given
f	current *sf* pattern
MF	current value of maximum flow
AVSSP	set of available *ssp*
TEMPSSP	temporary set of AVSSP
$sign(x)$	integer-valued function; +1 if $x > 0$, 0 if $x = 0$, and -1 if $x < 0$.

3.1 Algorithm

1. Initialize $f = (0,0,\cdots,0)$, MF = 0 and AVSSP = { all *ssp* };

2. TEMPSSP = AVSSP;

3. Select the shortest *ssp* P in TEMPSSP;
   if P is not an *fassp* **then**
   begin
   TEMPSSP = TEMPSSP $-$ $\{P\}$;
   if TEMPSSP = $\emptyset$ **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
   $f_i = f_i + sign(i) \cdot w$;
   if $|f_i| = c_i$ and i is a *udl* **then** AVSSP = AVSSP $-$ $\{P' \mid i \in P'\}$;
   end;
   Go to 2;

Example 1. Consider the bridge network in Figure 1, which has four *ssp*: (1,4), (2,5), (1,3,5), (2,-3,4) and all links except 3 are *udl*. Let the capacity vector be given as c = (2,6,2,5,3). The saturated *udl* are marked by '*s*' in *sf* patterns. The process stops when there is no more *fassp* left in AVSP, and the maximum flow of the network is computed as 7.

Table 1. Process for Figure 1

fassp	w	capacity vector	f	MF	AVSSP
-	-	(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) }

3.2 Mixed Networks

For a mixed network, we first find all *ssp* as being done for an undirected network. Note that the link direction of i in P does not depend on whether i is directed or not. Now, for a directed link, we call the pre-assigned direction of it in the network the *arc-direction*. A directed link is said to be a *reverse-arc in P*, if the arc-direction is opposite to the link direction of it in P. Since the flow cannot move reversely to the arc-direction on a directed link, we slightly change the definition of an *fassp* as follows.

Definition 3. For a mixed network, a *ssp* P is said to be a *fassp* with respect to f if it contains no saturated link in P and no flowless reverse-arc in P.

Example 2. For the bridge network presented in Figure 1, we suppose that all links are directed links in '+' direction. Superscripting a reverse-arc in a *ssp* with 'R', we have four *ssp*: (1,4), (2,5), (1,3,5), (2,-3^R,4), and the set of *udl* is {1,2,4,5}. Note that (2,-3^R,4) is no longer an *fassp* at the last step in Table 2, since it contains a flowless reverse-arc, i.e. -3^R.

Table 2. Process for directed bridge network

fassp	w	capacity vector	f	MF	AVSSP
-	-	(2,6,2,5,3)	(0,0,0,0,0)	0	{(1,4), (2,5), (1,3,5), (2,-3^R,4) }
(1,4)	2		(s,0,0,2,0)	2	{(2,5), (2,-3^R,4) }
(2,5)	3		(s,3,0,2,s)	5	{(2,-3^R,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, Aggarwal(1988), Lee and Park(2001), Rai and Soh(1991) and Varshney, Joshi and Chang(1994) for references. When a new subnetwork is formed by adding certain links to the existing subnetwork, Kyandoghere's(1998) method requires repeating the whole process all over again for the new subnetwork, since the residual capacity for each simple path in the new subnetwork may be changed. Rai and Soh(1991) also need to find all minimal cuts again and then, re-computes the sum of all link capacities for each of the minimal cuts for the new subnetwork. On the other hand, 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] Aggarwal, K.K. (1988). A fast algorithm for the performance index of a telecommunication network, *IEEE Transactions on Reliability*, Vol. 37, 65-69.

[2] Aggarwal, K.K. Chopra, Y.C. and Bajwa, J.S. (1982). Capacity consideration in reliability analysis of communication systems, *IEEE Transactions on Reliability*, Vol. 31, 177-181.

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

[4] Kyandoghere, K. (1998). A note on : reliability modeling & performance of variable link-capacity networks, *IEEE Transactions on Reliability*, Vol. 47, 44-48.

[5] Lee, S.M. and Park, D.H. (2001). An Efficient Method for Evaluation of Network Reliability with Variable Link-Capacities, *IEEE Transactions on Reliability*, Vol. 50, 374-379.

[6] Misra, K.B. and Prasad, P. (1982). Comments on: Reliability evaluation of a flow network, *IEEE Transactions on Reliability*, Vol. 31, 174-176.

[7] Rai, S. and Soh, S. (1991). A computer approach for reliability evaluation of telecommunication networks with heterogeneous link-capacities, *IEEE Transactions on Reliability*, Vol. 40, 441-451.

[8] Schanzer, R. (1995). Comment on: Reliability modeling and performance of variable link-capacity networks, *IEEE Transactions on Reliability*, Vol. 44, 620-621.

[9] Varshney, P.K., Joshi, A.R. and Chang, P.L. (1994). Reliability modeling and performance evaluation of variable link capacity networks, *IEEE Transactions on Reliability*, Vol. 43, 378-382.

[2002년 7월 접수, 2002년 10월 채택]