MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Computing a DAG using MA ordering for the all-to-one maximum flow routing problem

Yasuhiro OHARA and Shinji IMAHORI

METR 2007–09 February 2007

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/edu/course/mi/index_e.shtml

The METR technical reports are published as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Computing a DAG using MA ordering for the all-to-one maximum flow routing problem

Yasuhiro Ohara, Keio University
Shinji Imahori, University of Tokyo

Abstract

The purpose of the Internet routing system is to construct a Directed Acyclic Graph (DAG) for each destination. Although the number of paths and the connectivity of the graph are important properties, constructing a DAG that includes all edges to increase the number of paths and the connectivity has not been studied. In this paper we describe a new problem called all-to-one maximum flow routing problem that maximizes the minimum of maximum flow among all nodes to a destination. We studied an algorithm that utilizes the MA ordering to find the desirable DAG for a given network. It is proven that the algorithm produces the optimal solution, and the time complexity is the same with that of the MA ordering, $O(m + n \log n)$ where m and n are the number of edges and nodes, respectively. Simulations showed that the routing calculated by MA ordering outperforms current shortest path routing significantly in terms of maximum flow on each pair of nodes, but for the link utilization on a traffic demand of random model between multiple pair of nodes, MA ordering is outperformed by the shortest path routing because of a lack of efficient method to compute traffic split ratio.

Keywords: Algorithm, Multipath, Maximum flow problem

1 Introduction

The routing system in the Internet has a very important role since the path of all communication sessions in the Internet follow its decision. The path that the routing system decides determines the property of the communication session, such as availability, maximum amount of bandwidth and communication delay.

The Internet is a hop-by-hop network and all routers in the network forward packets autonomously, so they must agree on the paths to destinations in order to avoid routing loops. Such a configuration of routes is said to be a Directed Acyclic Graph (DAG) for the destination where routes to the nexthop in each routers indicate directed edges in the DAG.

The routing in the Internet has been based on the shortest path routing, in which each edge is given a cost for data to pass through, and the routing decision is made so that for each source and destination pair of nodes, the sum of the edge costs in the path is minimum. Since each router selects a shortest path to a destination, the routes to the destination form a tree rooted to the destination rather than a graph with multipaths to the destination. Although the IP routing architecture allows multipaths to the destination in a router where the path to the destination branches, multipaths to the destination in the shortest path routing require the costs to the destination on these paths to be equal, which rarely happen. Multipaths to the destination in the shortest path routing is called Equal Cost Multi Paths (ECMPs).

The tree based routing with very little multipaths has some deficiencies. First, if the shortest path includes malfunctioning routers or links on its way to the destination, all communications are not possible until the routing system detects the failure and recomputes the new shortest path. This means that the redundancy of the network graph can be utilized only after the routing change, and cannot be utilized totally if the routing system fails to detect the failure. This is against the belief that a dense and complex network graph structure exhibits reliability. Since making the network graph complex does not give advantages such as increased reliability, there are still many tree-based network graph structures in the Internet.

Second, the maximum available bandwidth between the source and the destination is limited to that of the shortest path. If the transmission rate exceeds the maximum available bandwidth of the shortest path, the shortest path congests without using alternative roundabout paths with available bandwidth even if they exist. It means that the network cannot support traffic demands that have an excessive amount of traffic on a source-destination pair compared to the shortest path, limiting the range of the maximum traffic supported by the network.

We propose a novel multipath route calculation algorithm that construct a DAG that includes all edges in the graph in this paper. A new graph problem, *all-to-one maximum flow routing problem*, has been introduced to describe the problem solved by our algorithm. Our algorithm utilizes MA ordering [7, 8] to optimally solve the problem. The main contributions of this paper are: (a) introducing a graph problem with new objective function; (b) proposing a novel route calculation algorithm for the problem; (c) verifying, for the first time, the tolerance property of the routing with maximum multipaths against link congestion with a random model traffic demand.

The rest of this paper is organized as follows. Section 2 presents the related work of multipath routing algorithms. Section 3 introduces the all-to-one maximum flow routing problem to formulate the problem. Section 4 describes our algorithm that utilizes MA ordering. Section 5 gives the proof of correctness of our algorithm. Section 6 presents some simulations to evaluate the property of the algorithm. We conclude in Section 7.

2 Related Work

Proposed multipath routing methods are based on the shortest path routing and are essentially its extensions. MPDA [11], MDVA [12] and MPATH [10] use the Loop Free Invariant (LFI) property of shortest path routing to compute multipath routes. FIR [5] computes per network interface routing table by executing Shortest Path First (SPF) calculations the number of times equal to the number of neighbors for each router to route around the failure between the router and the neighbor. Deflection [13] extends the LFI property utilizing the identity of the incoming network interface to produce increased successor set.

3 Problem Description

We are given an undirected graph and a destination node. We choose some edges from the given edges and decide directions of the chosen edges. To avoid routing loops, the resulting directed graph should not have any directed cycle; i.e., we want to find a DAG on the input graph. For the purpose of reliability and robustness of the network, we consider the edge connectivity from all nodes to the destination node. By maximizing the minimum connectivity among all nodes to the destination, we obtain a routing that achieves the most robustness. By calculating such routing for each destination, we obtain the routing that achieves the most robustness for all source-destination node pairs. We call the problem of deciding the directions of edges to maximize the minimum connectivity among all nodes to the destination the *all-to-one maximum connectivity routing problem*. The formulation of this problem is obtained by setting the capacities of all edges as 1 in the all-to-one maximum flow routing problem introduced below.

In order to support a traffic demand that has an excessive amount of traffic on a source-destination pair compared to the shortest path, we consider the maximum flow from every node to the destination node. By maximizing the minimum maximum flow among all nodes to the destination, we obtain a routing that achieves the best support for the excessive amount of traffic to the destination. By calculating such routing for every destination, we obtain the routing that achieves the best support for all source-destination node pairs. We call the problem of maximizing the minimum maximum flow among all nodes to the destination the *all-to-one maximum flow routing problem*.

The formulation of the all-to-one maximum flow routing problem is as follows. We are given a capacitated undirected graph $G = (V, E, cap)$. Let $n = |V|$ and $m = |E|$, and $cap(v, u)$ gives the capacity of the edge (v, u) if $(v, u) \in E$, otherwise 0. We are also given a node $t \in V$ as the destination. For each orientation p of the edges, we denote by $f_p(v, t)$ the amount of maximum flow from v to t. (We assume $f_p(t,t) = +\infty$ for every orientation p.) The problem is to find an orientation of the given graph that maximizes the minimum maximum flow among all nodes to the destination t under the condition that the resulting graph is acyclic.

Maximize:

$$\min_{v \in V} f_p(v,t).$$

4 Algorithm

We design an algorithm for the all-to-one maximum flow routing problem. Instead of deciding the direction of each edge explicitly, we determine a permutation of nodes (i.e., nodes are labeled from 1 to n), and set the direction of each edge from higher-labeled node to lower-labeled node. It is known that constructing a DAG on an undirected graph is equivalent to deciding a topological order of nodes [1]. We propose to use the maximum adjacency (MA) ordering for solving the problem.

First, we explain the MA ordering proposed by Nagamochi and Ibaraki [7]. Let $G = (V, E)$ be an undirected graph that has n nodes and m edges. An ordering $v_1, v_2, \ldots, v_n$ of nodes is called an MA ordering if an arbitrary node is chosen as v_1, and after choosing the first i nodes $v_1, v_2, \ldots, v_i$, the $(i+1)$-st node v_{i+1} is chosen from the nodes v that have the largest number of edges between v and $\{v_1, \ldots, v_i\}$. It is known that the MA ordering is useful for various problems on graphs such as to identify a minimum cut between two nodes and to solve the edge-connectivity augmentation problem. An algorithm to compute an MA ordering is given in Algorithm 1, where $d(v, S)$ denotes the number of edges between a node v and a set of nodes S.

Algorithm 1 MA ordering algorithm

1: **procedure** MA ORDERING($G = (V, E), s \in V$)
2:     $v_1 \leftarrow s,\ S = \{s\},\ T = V \setminus \{s\}$
3:     $i \leftarrow 2$
4:     **while** $i \leq |V|$ **do**
5:         choose a node $v \in T$ with the largest $d(v, S)$
6:         $v_i \leftarrow v,\ S = S \cup \{v\},\ T = T \setminus \{v\}$
7:         $i \leftarrow i + 1$
8:     **end while**
9:     output ordering $(v_1, v_2, \ldots, v_n)$ of nodes
10: **end procedure**

For a capacitated undirected graph $G = (V, E, cap)$, an ordering similar to the MA ordering is also defined. In this case, we choose a node $v \in T$ in Line 5 of Algorithm 1 as the maximum $\sum_{u \in S} cap(v, u)$ instead of $d(v, S)$. By using an appropriate data structure such as Fibonacci heap [4], an MA

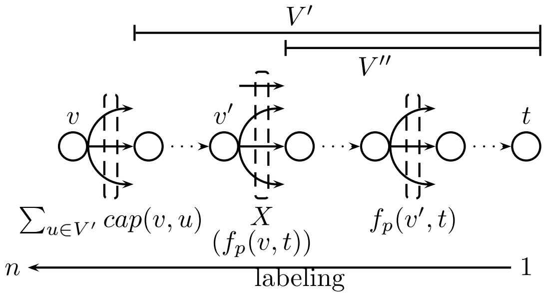

Figure 1: The relation between v, V', v', V'', X and t in Lemma 5.1

ordering for a (capacitated) undirected graph and for a node $s \in V$ can be obtained in $\mathrm{O}(m + n \log n)$ time [7].

Now, we propose an algorithm for the all-to-one maximum flow routing problem. Our algorithm is very simple: We compute an MA ordering for a capacitated undirected graph $G = (V, E, cap)$ using a destination node $t \in V$ as an initial node s. We then output the direction of each edge from the higher-labeled node to the lower-labeled node. This algorithm runs in $\mathrm{O}(m + n \log n)$ time. We show the optimality of our algorithm in the next section.

5 Correctness of our algorithm

In this section, we give a proof for the optimality of our algorithm proposed in the previous section. In other words, we show that an MA ordering of nodes gives an optimal solution for the all-to-one maximum flow routing problem.

Let a capacitated undirected graph $G = (V, E, cap)$ and a destination node $t \in V$ be given, where the graph has n nodes including the destination node. We first show the following lemma on the minimum cut and the maximum flow for a bottleneck node, when an orientation p of edges is given with a permutation of nodes; i.e., nodes are labeled from 1 to n and each edge is headed to the lower-labeled node.

Lemma 5.1 *Let v be the lowest-labeled node with the minimum $f_p(v,t)$. Then, $f_p(v,t) = \sum_{u \in V'} cap(v,u)$ holds, where V' is the set of nodes which have lower-labels than v.*

Proof. By using a relationship between the cut and flow on a DAG, $f_p(u',t) \leq \sum_u cap(u',u)$ holds for every node $u' \neq t$, where u has a lower label than u'.

We prove the equality by contradiction. Suppose that $f_p(v,t) < \sum_{u \in V'} cap(v,u)$ holds. The max-flow min-cut theorem on a directed graph [2] implies that there exists a directed cut X (i.e., a partition of nodes) whose value is equal to $f_p(v,t)$. Let v' is the lowest-labeled node such that v and v' belong to the same subset of nodes partitioned by X. Then, the following inequalities hold:

$$f_p(v,t) \geq \sum_{u \in V''} cap(v',u) \geq f_p(v',t)$$

where V'' is the set of nodes which have lower-labels than v'. (The relation between v, V', v', V'', X, and t is shown in Figure 1.) This contradicts the assumption that v is the lowest-labeled node with the minimum maximum flow to t. □

Theorem 5.2 *The MA ordering solves the all-to-one maximum flow routing problem optimally.*

Proof. Suppose that the MA ordering algorithm using the destination node t as an initial node gives a label i to node v_i for every node in V (i.e., we assume the destination node $t = v_1$ and it has a label 1). We set the direction of each edge from the higher-labeled node to the lower-labeled node. We call this orientation "*ma*," and let g_{ma} be the objective value for this orientation. We define $V_i = \{v_1, v_2, \ldots, v_{i-1}\}$ as the set of nodes which have smaller labels than v_i. Let $c(v, V')$ denote $\sum_{u \in V'} cap(v,u)$. By the definition of $c(v,V')$, $c(v,B) \leq c(v,A)$ if $B \subseteq A$ holds.

Let v_k be the lowest-labeled node whose maximum flow value $f_{ma}(v_k,t)$ to the destination node t is the smallest. By using Lemma 5.1, we have the following equalities:

$$g_{ma} = f_{ma}(v_k,t) = c(v_k, V_k).$$

From the MA ordering algorithm, $c(v_i, V_k) \leq c(v_k, V_k)$ holds for $i = k+1, k+2, \ldots, n$, because v_k is labeled earlier than v_i.

We assume that there exists another ordering of nodes, whose orientation is denoted by "*opt*" and $g_{opt} > g_{ma}$ holds. Let v_l be the node with the smallest label in *opt* among a set of nodes $\{v_k, v_{k+1}, \ldots, v_n\}$. This is illustrated in Figure 2. Then, the following equalities and inequalities hold:

$$g_{opt} \leq f_{opt}(v_l,t) \leq c(v_l, V') \leq c(v_l, V_k) \leq c(v_k, V_k) = g_{ma},$$

where V' is the set of nodes which have smaller labels than v_l in *opt*. This contradicts the assumption that there exists an orientation *opt* with $g_{opt} > g_{ma}$. □

$$f_{ma}(v_k, t) = g_{ma}$$

$$\begin{array}{ll}
 & \qquad\qquad\qquad\qquad\quad V_k \\
ma & v_n, \ldots, v_l, \ldots, v_k, \overleftrightarrow{\ldots, \ldots, v_1}(= t) \\
opt & v_x, \ldots, v_k, \ldots, \ldots, v_l, \underleftrightarrow{\ldots, v_1}(= t) \\
 & \qquad\qquad\qquad\qquad\quad\; V'
\end{array}$$

Figure 2: The optimality proof of MA ordering

6 Evaluation

We evaluate our method in three aspects; robustness (Section 6.1), maximum flow amount on a pair of nodes (Section 6.3), and link utilization (Section 6.4).

6.1 Robustness

Constructing a DAG that includes all edges in the graph increases the redundancy of paths and the robustness of the network. Our method is used by Ohara et al. [9] along with some extensions to the IP architecture for the redundancy purpose, and showed the improved failure avoidance property of our method compared to a recent existing method [13].

6.2 Simulation methodology

We ran two simulations to show the property of our method. One is to show that our method increases the maximum flow amount for each source-destination node pair. The other is to show how the routing derived from our method exhibits the link utilization property in the network with random model traffic demands between multiple pair of nodes.

A virtual topology generated by BRITE [6] is used in the simulations. The configurations of the BRITE are as follows:

1. Topology generation model is the Barabási-Albert model.
2. Number of nodes in the graph (N) is 20.
3. The node placement is random in the plane size HS = 1,000 and LS = 100.
4. The number of neighboring nodes each new node connects (m) is 4.
5. The distribution of bandwidth in links (BWDist) are uniform in the range between 100 and 1,000.

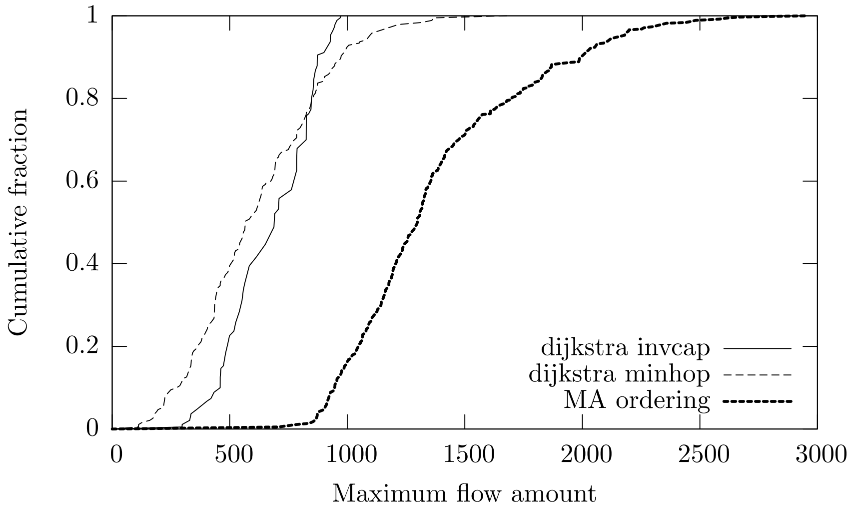

Figure 3: The maximum flow amount per source-destination node pair

We compared our method to the classic Dijkstra's algorithm with two cost settings: the inverse capacity and the minimum hop. Dijkstra's algorithm with inverse capacity cost setting is called "dijkstra invcap," and the cost of each link is configured to the value that is inversely proportional to the bandwidth generated by BRITE. The edge's cost c is derived by the equation $c = 100,000/\text{bandwidth}$. Dijkstra's algorithm with minimum hop cost setting is called "dijkstra minhop," and the costs of all links are configured as 1. In both cost settings, the Equal Cost Multi Paths (ECMPs) are properly calculated and considered in the simulation results. Our method is called "MA ordering."

6.3 Maximum flow amount on a pair of nodes

The Cumulative Distribution Function (CDF) of maximum flow amount in all source-destination node pairs is given in Figure 3. As is proven in Section 5, our method calculates the routing for hop-by-hop network that maximize the minimum maximum flow amount among all source-destination node pairs. Our method outperforms the other two significantly. 84.2% of the source-destination pairs have more than 1,000 maximum available bandwidth in MA ordering, while in dijkstra minhop it is only 7.6% and in dijkstra invcap it is 0%.

6.4 Link utilization

We also examined the link utilization with traffic demands on multiple source-destination node pairs. The traffic model is drawn from the previous work of Fortz and Thorup [3]. It is a random model with two notions,

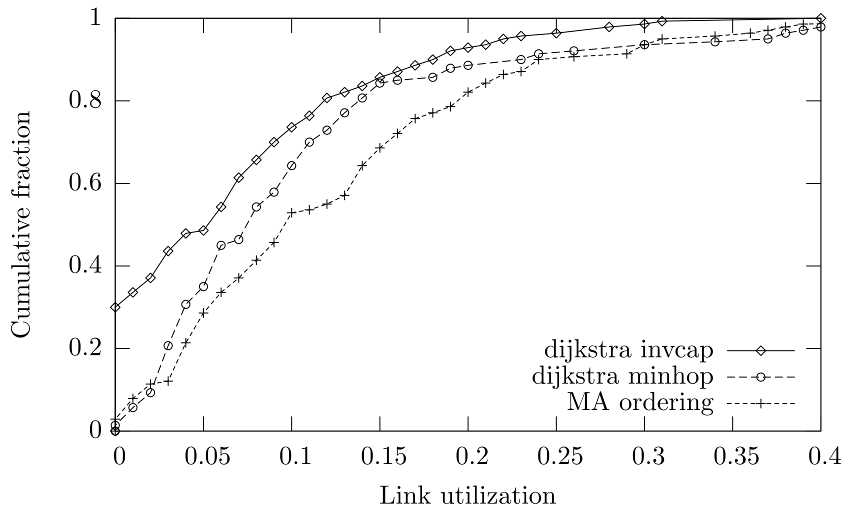

Figure 4: The link utilization for each link

one models hot spot node that sends and receives more data, and the other models relatively more demand between pair of nodes that are close in terms of Euclidean distance generated by BRITE. The scaling parameter α is set to 100. The result of loading a traffic demand derived from the model on the network is present in Figure 4. Our method of using MA ordering is outperformed by the two classical routing methods significantly.

The reason for this is closely related to the traffic splitting ratio in each node. In this simulation, we split the traffic that the node is forwarding, in the ratio of the outgoing edge's bandwidth to the sum of bandwidth of all outgoing edges from the node to the destination. This method does not consider the relation between the amount of traffic that the node splits and the link bandwidth capacity that the traffic will pass through. Consequently, in terms of link utilization, this method is outperformed by the other methods that are simple and classical. For the same reason, MA ordering is also outperformed, in terms of the maximum link utilization among all links, by dijkstra invcap. Each maximum link utilization in the network is 0.398 in dijkstra invcap, 0.521 in dijkstra minhop, and 0.507 in MA ordering.

Generally the method of constructing a DAG that includes all edges in the graph calculates significantly longer paths that are roundabout. If the paths are roundabout, the efficiency of using link bandwidth against the amount of traffic load decreases, since the same amount of traffic is forwarded in longer path consuming additional link bandwidth.

These results can be considered as either trade-off for the increased robustness of the network, or simply the non-optimal settings of the traffic splitting ratio. Improving the settings of the traffic splitting ratio in the multipath routing calculated by our method is necessary, and we consider it as the future work.

7 Conclusion

In this paper, we presented a novel edge orientation problem "*all-to-one maximum flow routing problem*," in which the objective is to maximize the minimum maximum flow amount from all nodes to the destination. We introduced the application of MA ordering to an edge orientation problem, and this application is proved to find an optimal ordering of nodes (thus an optimal DAG) for the purpose.

Multipath routing has been researched for long time, but this paper studied for the first time an approach to construct a DAG that includes all edges in the network graph. The lack of efficient method to compute the traffic split ratio on each node is found to be problematic when our method is applied in the communication networks as multipath routing. In order to gain the robustness property our method provides, the method to compute the traffic split ratio on a DAG needs to be developed.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[2] G. B. Dantzig and D. R. Fulkerson. *On the max flow min cut theorem of networks.* The RAND Corporation, Santa Monica, California, 1955. Paper P-826.

[3] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *INFOCOM*, pages 519–528, 2000.

[4] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, 1987.

[5] S. Lee, Y. Yu, S. Nelakuditi, Z.-L. Zhang, and C.-N. Chuah. Proactive vs reactive approaches to failure resilient routing. In *INFOCOM*, 2004.

[6] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An approach to universal topology generation. In *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS '01*, August 2001.

[7] H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM J. Discret. Math.*, 5(1):54–66, 1992.

[8] H. Nagamochi and T. Ibaraki. Graph connectivity and its augmentation: applications of MA orderings. *Discrete Appl. Math.*, 123(1-3):447–472, 2002.

[9] Y. Ohara, H. Kusumoto, O. Nakamura, and J. Murai. Routing architecture for dependable internet: User-driven routing to bypass failures. *IEICE Transactions on Communications*, To be published, 2007.

[10] S. Vutukury and J. J. Garcia-Luna-Aceves. An algorithm for multipath computation using distance-vectors with predecessor information. In *IEEE Computer Communications and Networks, 1999. Proceedings. Eight International Conference on*, pages 534–539, Oct 1999.

[11] S. Vutukury and J. J. Garcia-Luna-Aceves. A simple approximation to minimum-delay routing. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 227–238, New York, NY, USA, 1999. ACM Press.

[12] S. Vutukury and J. J. Garcia-Luna-Aceves. MDVA: A distance-vector multipath routing protocol. In *INFOCOM*, pages 557–564, 2001.

[13] X. Yang and D. Wetherall. Source selectable path diversity via routing deflections. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 159–170, New York, NY, USA, 2006. ACM Press.