

# **MATHEMATICAL ENGINEERING TECHNICAL REPORTS**

## **Improved Max-Flow Min-Cut Algorithms in a Circular Disk Failure Model with Application to a Road Network**

Kensuke OTSUKI, Yusuke KOBAYASHI,  
and Kazuo MUROTA

METR 2015-13

March 2015

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.keisu.t.u-tokyo.ac.jp/research/techrep/index.html>

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.

# Improved Max-Flow Min-Cut Algorithms in a Circular Disk Failure Model with Application to a Road Network\*

kensuke OTSUKI      Yusuke KOBAYASHI      Kazuo MUROTA

Department of Mathematical Informatics  
Graduate School of Information Science and Technology  
The University of Tokyo  
kensuke\_otsuki@mist.i.u-tokyo.ac.jp  
kobayashi@mist.i.u-tokyo.ac.jp  
murota@mist.i.u-tokyo.ac.jp

March 2015

## Abstract

In the evaluation of network reliability, the objectives are to model reliability of networks appropriately and to compute it in a realistic time. We can consider various models of reliability of networks, and Bienstock (1991) first introduced a geographical failure model where each failure is represented by a 2-dimensional region. In this model, we consider the situation that geographical networks such as road networks may be damaged by externally caused disasters, and such disasters may destroy several links of the networks simultaneously, rather than each link independently. Recently, Neumayer–Efrat–Modiano (2012) investigated the max-flow problem and the min-cut problem under the Circular Disk Failure Model, in which the shape of each failure is restricted to be a disk. Under this model, Kobayashi–Otsuki (2014) gave polynomial time algorithms to find optimal solutions of these two problems. In this paper, we improve the algorithms and evaluate their performance by computational experiments. Owing to the improvements, we can find in a realistic time optimal solutions of the max-flow problem and the min-cut problem in large networks under the Circular Disk Failure Model. As a realistic instance, we analyze reliability of a road network in NewYork consisting of 264,346 nodes.

## 1 Introduction

Recently, with the development of various networks such as road networks and the Internet, there is urgent need to evaluate reliability of large networks. In the evaluation of reliability, the crucial points are how to model reliability of networks appropriately and how to compute it in a realistic time. We can conceive various ways to formulate reliability of networks (see e.g. [4]), but a complicated formulation of reliability would make it hard to design an efficient algorithm for computing reliability.

---

\*This work was partially supported by JST, ERATO, Kawarabayashi Large Graph Project, Japan. The second and third authors are supported by Grant-in-Aid for Science Research, MEXT, Japan.

In most previous studies, connectivity of networks is regarded as one of the most significant factors of network reliability, where connectivity is defined as the minimum number of link-failures (or node-failures) that disconnect the network. Connectivity is a simple and natural model for reliability of networks, and an advantage of using connectivity is that we have efficient algorithms for computing it (see [10]). In particular, there are many studies on the problem of finding a minimum number of link-failures that disconnect two specified nodes  $s$  and  $t$  (which we call the min-cut problem) and its dual problem called the max-flow problem. It is known as the max-flow min-cut theorem [3, 6] that the optimal values of the min-cut problem and the max-flow problem are equal, and the proof of this theorem leads to algorithms for these two problems. Since then, many faster algorithms for the max-flow problem and the min-cut problem have been proposed (see [10]), and currently we can solve these problems for huge networks in a realistic time.

In this paper, we consider variants of the max-flow problem and the min-cut problem, motivated by modeling geographical failures on geographical networks such as road networks. Geographical networks may be damaged by externally caused disasters like earthquakes or floods. Such disasters, occurring in specific geographical regions, may often destroy several links of the networks simultaneously, rather than each link independently. In 1991, Bienstock [2] first introduced a geographical failure model where each failure is represented by a 2-dimensional region. Recently, various models which takes such region failures into consideration have been drawing a great deal of attention [1, 8, 9, 11, 12]. In particular, Neumayer–Efrat–Modiano [7] adopted the *Circular Disk Failure Model* for the max-flow problem and the min-cut problem, where each failure is represented by a disk with a predetermined size. This model can be regarded as a special case of Bienstock’s model, in which the shape of each failure is restricted to be a disk. Under that model, the min-cut problem is to find a minimum number of disk failures that disconnect two specified nodes  $s$  and  $t$  of the given planar graph  $G$ , and the max-flow problem is to find a maximum number of  $s$ - $t$  paths such that no disk failure intersects a pair of these paths.

Under the Circular Disk Failure Model, Neumayer–Efrat–Modiano [7] gave a polynomial time min-cut algorithm and Kobayashi–Otsuki [5] gave a polynomial time max-flow algorithm. A faster and simpler min-cut algorithm was also given in [5]. However, as we can see in [13], many of real road networks consist of more than 100,000 nodes, which are too huge for the aforementioned algorithms to solve in a realistic time.

In this paper, we improve the max-flow algorithm and the min-cut algorithm given in [5], and evaluate the performance of these algorithms by computational experiments. Owing to the improvements, now we can find in a realistic time optimal solutions of the max-flow problem and the min-cut problem under the Circular Disk Failure Model in huge networks consisting of more than 100,000 nodes. As a practical instance, we analyze reliability of a road network in NewYork [13] with 264,346 nodes. We can solve the problems in the network in about 30 seconds on a computer with Intel Core i7, 2.8 GHz and 8 GB of memory, while it takes about 3 hours by the existing algorithms.

The remaining of this paper is organized as follows. In Section 2, we give formal descriptions of the problems with some mathematical definitions. In Sections 3 and 4, we describe our improvements of the max-flow algorithm and the min-cut algorithm, respectively. In Section 5, we evaluate the performance of our algorithms and apply our algorithms to the road network in NewYork. Finally, in Section 6, we conclude this paper.

## 2 Problem Settings

Let  $G = (V, E)$  be a graph drawn in the plane with a node set  $V$ , a link set  $E$ , and two distinct nodes  $s, t \in V$ . Throughout this paper we assume that each link is drawn as a line segment, and a

link is sometimes called an edge. Let  $r_B$  be a hole radius, where each hole represents a failure of the network. We consider two disks of another radius  $r_P$  whose centers are  $s$  and  $t$ , respectively. We call them *protective disks* and their radius  $r_P$  *protective radius*. Define  $\mathcal{H}(r_B, r_P)$  as the set of all disks of radius  $r_B$  whose centers are not contained in the protective disks of radius  $r_P$ . We call each element of  $\mathcal{H}(r_B, r_P)$  a *hole* in this paper. We shall think of each hole as a closed set, including its boundary.

The geographical min-cut problem and the geographical max-flow problem introduced by Neumayer–Efrat–Modiano [7] are defined as below. Solutions of the problems represent connectivity of the graph between two nodes  $s$  and  $t$  under the Circular Disk Failure Model.

#### Geographical Min-Cut by Circular Disasters (GMCCD)

**Input:** a graph  $G = (V, E)$  drawn in the plane, two distinct nodes  $s$  and  $t$ , a hole radius  $r_B$ , and a protection radius  $r_P(> r_B)$ .

**Find:** a minimum cardinality set of holes in  $\mathcal{H}(r_B, r_P)$  that disconnect  $s$  from  $t$ .

#### Geographical Max-Flow by Circular Disasters (GMFCD)

**Input:** a graph  $G = (V, E)$  drawn in the plane, two distinct nodes  $s$  and  $t$ , a hole radius  $r_B$ , and a protection radius  $r_P(> r_B)$ .

**Find:** a maximum cardinality set of  $s$ - $t$  paths such that no hole in  $\mathcal{H}(r_B, r_P)$  intersects a pair of these paths.

In what follows, we say that two  $s$ - $t$  paths are *hole-disjoint* if there is no hole in  $\mathcal{H}(r_B, r_P)$  that intersects both of them. Let MIN-CUT and MAX-FLOW denote the optimal values of the GMCCD and the GMFCD, respectively, and we call feasible solutions of the GMCCD and the GMFCD a *hole cut* and a *hole-disjoint flow*, respectively. The reason we introduce protective disks is that the GMCCD and the GMFCD will have trivial optimal solutions if a hole contains  $s$  or  $t$ . In terms of application, we can assume that  $s$  and  $t$  are well-protected so that no failure can occur nearby nodes  $s$  and  $t$ .

**Example 1.** Consider the graph as in Figure 1. Small circles represent holes of radius  $r_B$  and two large circles represent protective disks of radius  $r_P$ . In the graph, we can easily see that MAX-FLOW = 1 and MIN-CUT = 2. The thick  $s$ - $t$  path represents an optimal solution of the GMFCD, and two small circles represent an optimal solution of the GMCCD.

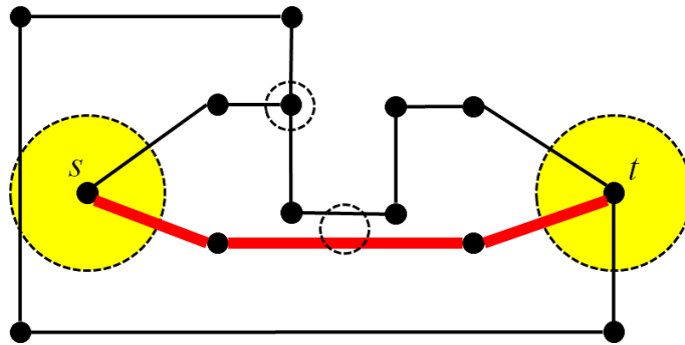


Figure 1: Example of the problems

This example is given in [7] as an example to show that  $\text{MIN-CUT} = \text{MAX-FLOW}$ , which is an analogy to Menger's theorem, does not necessarily hold. Practically, however,  $\text{MIN-CUT} = \text{MAX-FLOW}$  holds in most cases. Indeed, it is claimed in [7] that among 1000 randomly generated instances with 13 nodes,  $\text{MIN-CUT} = \text{MAX-FLOW}$  holds for 99% of the instances and  $\text{MIN-CUT} = \text{MAX-FLOW} + 1$  holds for the remaining 1%. Later, it is proved in [5] that  $\text{MAX-FLOW} \leq \text{MIN-CUT} \leq \text{MAX-FLOW} + 1$  holds for any instances.

In this paper, we improve the existing polynomial time algorithms [5] for the GMCCD and the GMFCD under the assumption that  $\text{MIN-CUT} = \text{MAX-FLOW}$ . Although we need the extra assumption that  $\text{MIN-CUT} = \text{MAX-FLOW}$  for our improvements, this equality holds in almost all practical cases. Indeed,  $\text{MIN-CUT} = \text{MAX-FLOW}$  holds in a large practical network considered in Section 5, which motivates us to focus on the case meeting this assumption.

Here we introduce some mathematical definitions concerning the GMCCD and the GMFCD, following [5]. For any  $s$ - $t$  path  $P$ , we think of  $P$  as being oriented from  $s$  to  $t$ , and denote by  $P^{-1}$  the path oriented from  $t$  to  $s$  along  $s$ - $t$  path  $P$ . For two  $s$ - $t$  paths  $P'$  and  $P''$  without crossings, let  $R(P', P'')$  denote the closed region encircled by  $P' \cdot (P'')^{-1}$  in clockwise orientation, where  $P' \cdot (P'')^{-1}$  is the closed curve obtained by concatenating  $P'$  and  $(P'')^{-1}$ . Let  $|R(P', P'')|$  be the number of faces of  $G$  contained in  $R(P', P'')$ . For two  $s$ - $t$  paths  $P', P''$  without crossings, we call a pair  $(P', P'')$  *clockwise hole-disjoint* if for any hole  $H \in \mathcal{H}(r_B, r_P)$ ,  $R(P', P'') - H$  is connected. Note that if two  $s$ - $t$  paths  $P'$  and  $P''$  are hole-disjoint,  $(P', P'')$  is clockwise hole-disjoint, while the opposite is not always true.  $P'$  and  $P''$  are hole-disjoint if and only if both pairs  $(P', P'')$  and  $(P'', P')$  are clockwise hole-disjoint.

For a pair of edges  $e, e' \in E$ , if there exists a hole  $H \in \mathcal{H}(r_B, r_P)$  that intersects both edges, we take two points  $w_{e,e'}$  on  $e$  and  $w_{e',e}$  on  $e'$  arbitrarily that are contained in hole  $H$ . Then, define

$$W := \{w_{e,e'}, w_{e',e} \mid e, e' \in E \text{ contained in a common hole}\}.$$

Theoretically,  $|W|$  is obviously bounded by  $|E|^2$ . In practical networks,  $|W| = O(|E|)$ , that is,  $|W| = O(|V|)$ , since there are only a small number of pairs that are close enough to be contained in a common hole in  $\mathcal{H}(r_B, r_P)$ .

### 3 Max-Flow Algorithms

#### 3.1 Existing Max-Flow Algorithm

In this subsection, we review the algorithm for the GMFCD given by Kobayashi–Otsuki [5]. Algorithm 1 either finds pairwise hole-disjoint  $k+1$   $s$ - $t$  paths or reports the non-existence of such  $s$ - $t$  paths, under the assumption that we already have pairwise hole-disjoint  $k$   $s$ - $t$  paths.

**Lemma 1** ([5]). *Suppose that we are given a graph  $G = (V, E)$  drawn in the plane, two distinct nodes  $s$  and  $t$ , a hole radius  $r_B$ , and a protection radius  $r_P (> r_B)$ . Moreover suppose that we have pairwise hole-disjoint  $k$   $s$ - $t$  paths  $P_1, \dots, P_k$ , and an  $s$ - $t$  path  $P_{k+1}$  in  $R(P_k, P_1)$  such that  $(P_k, P_{k+1})$  is clockwise hole-disjoint. If Algorithm 1 reports the non-existence of pairwise hole-disjoint  $k$   $s$ - $t$  paths, then  $\text{MAX-FLOW} = k$ .*

An example of iterations in Algorithm 1 is given in Figure 2. Now, by using Algorithm 1 with  $k = 1, 2, \dots$  in order, an optimal solution of the GMFCD can be computed in polynomial time (see Algorithm 2).

**Theorem 2** ([5]). *An optimal solution of the GMFCD can be computed in polynomial time.*

*Proof.* Lemma 1 implies that an optimal solution of the GMFCD can be computed by using Algorithm 1 with  $k = 1, 2, \dots$  in order at most  $|V|$  times (see Algorithm 2). For each  $k$ , the number of iterations in Algorithm 1 is at most  $|W|$ . Since each iteration runs in  $O(|W|)$  time, the total running time is  $O(k|W|^2)$  ( $k := \text{MAX-FLOW}$ ), which is polynomial.  $\square$

As pointed out in [5], in most practical cases, the running time is  $O(|V|^2)$  since  $k$  is small and  $|W| = O(|V|)$ .

---

**Algorithm 1** Induction step

---

**Input:** pairwise hole-disjoint  $k$   $s$ - $t$  paths  $P_1, \dots, P_k$  and an  $s$ - $t$  path  $P_{k+1}$  in  $R(P_k, P_1)$  such that  $(P_k, P_{k+1})$  is clockwise hole-disjoint

**Output:** pairwise hole-disjoint  $k + 1$   $s$ - $t$  paths

- 1: **for**  $l = k + 1, k + 2, \dots, k + |W| + 2$  **do**
  - 2:   **if**  $(P_l, P_{l-k})$  is clockwise hole-disjoint **then**
  - 3:     **return**  $P_{l-k}, \dots, P_{l-1}, P_l$ , which are pairwise hole-disjoint paths
  - 4:   **else**
  - 5:     let  $P_{l+1}$  be the  $s$ - $t$  path in  $R(P_{l-k}, P_{l-k+1})$  such that  $(P_l, P_{l+1})$  is clockwise hole-disjoint and  $R(P_{l+1}, P_{l-k+1})$  is maximized under this condition
  - 6:   **end if**
  - 7: **end for**
  - 8: reports the non-existence of pairwise hole-disjoint  $k + 1$   $s$ - $t$  paths
- 

---

**Algorithm 2** Hole-disjoint max-flow algorithm

---

**Output:** a maximum cardinality set of pairwise hole-disjoint  $s$ - $t$  paths

- 1: **if** two nodes  $s, t$  are not connected **then**
  - 2:   **return**  $\emptyset$  ( $\text{MAX-FLOW} = 0$ )
  - 3: **else**
  - 4:   find an  $s$ - $t$  path  $P_1$
  - 5:   **for**  $l = 1, 2, \dots, |V|$  **do**
  - 6:     let  $P_{l+1}$  be an  $s$ - $t$  path in  $R(P_l, P_1)$  such that  $(P_l, P_{l+1})$  is clockwise hole-disjoint
  - 7:     apply Algorithm 1 with  $k = l$
  - 8:     **if** Algorithm 1 reports the non-existence of pairwise hole-disjoint  $k + 1$   $s$ - $t$  paths **then**
  - 9:       **return** pairwise hole-disjoint  $l$   $s$ - $t$  paths ( $\text{MAX-FLOW} = l$ )
  - 10:    **else**
  - 11:     obtain pairwise hole-disjoint  $l + 1$   $s$ - $t$  paths  $P_1, \dots, P_{l+1}$
  - 12:    **end if**
  - 13: **end for**
  - 14: **end if**
- 

### 3.2 Improvement of Max-Flow Algorithm

We improve the max-flow algorithm (Algorithm 2) of [5] in this subsection. As we reviewed in the previous subsection, the running time of the algorithm is  $O(|V|^2)$  practically. The improvement does not change the theoretical computational time complexity; in other words, it still takes  $O(|V|^2)$  time to solve the GMFCD in the worst case. Practically, however, the running time of the algorithm is decreased drastically in most cases, as we will see in Section 5.2.

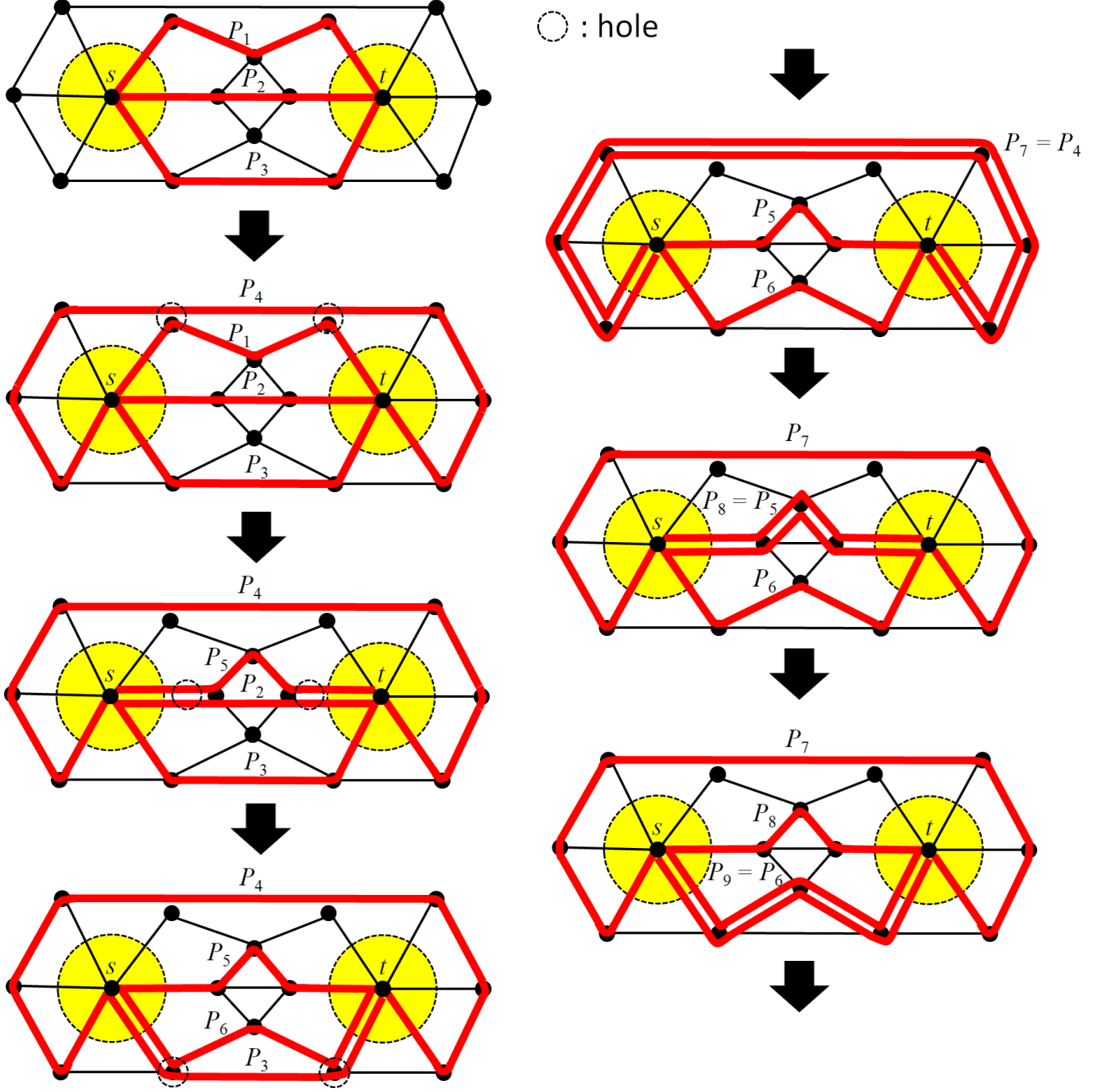


Figure 2: Example of iterations in Algorithm 1 ( $k = 3$ ). We begin with pairwise hole-disjoint three  $s$ - $t$  paths  $P_1, P_2, P_3$ , and another path  $P_4$  lying in  $R(P_3, P_1)$  such that  $(P_3, P_4)$  is clockwise hole-disjoint. In this example,  $\text{MAX-FLOW} = 3$ , and so Algorithm 1 reports the non-existence of pairwise hole-disjoint four  $s$ - $t$  paths after finite iterations. This figure shows the first five iterations. Note that  $P_7$  and  $P_4$  coincide with each other, and similarly  $P_8 = P_5$  and  $P_9 = P_6$  hold. Moreover we can easily see that  $P_{l+3} = P_l$  holds for  $l \geq 4$ .



Recall that Algorithm 2 uses Algorithm 1 with  $k = 1, 2, \dots$  in order, where Algorithm 1 either finds pairwise hole-disjoint  $k + 1$   $s$ - $t$  paths or reports the non-existence of such  $s$ - $t$  paths, under the assumption that we already have pairwise hole-disjoint  $k$   $s$ - $t$  paths  $P_1, \dots, P_k$ .

Let  $k' := \text{MAX-FLOW}$ . The most time-consuming part of Algorithm 2 is Algorithm 1 with  $k = k'$ , which needs  $|W|$  iterations, whereas the total running time of Algorithm 1 with  $k = 1, 2, \dots, k' - 1$  is much smaller since only a small number of iterations are necessary. Therefore, to improve the running time of Algorithm 2, we have to aim at improving Algorithm 1 with  $k = k'$  so that it needs a smaller number of iterations.

Suppose that we have pairwise hole-disjoint  $k'$   $s$ - $t$  paths  $P_1, \dots, P_{k'}$ . We note that it is sufficient to retain only  $k$   $s$ - $t$  paths  $P_{l-k+1}, \dots, P_{l-1}, P_l$  in the  $l$ -th iteration in Algorithm 1 with  $k = k'$ . Then, we show that if  $P_{l+1} = P_{l-k+1}$  holds in some  $k$  consecutive iterations indexed by  $l$ , it will hold in the following iterations (Theorem 3 below). For example, in Figure 2,  $P_{l+3} = P_l$  holds for  $l \geq 4$ .

This fact shows that if  $P_{l+1} = P_{l-k+1}$  holds in some  $k$  consecutive iterations in Algorithm 1, we can break iterations and terminate Algorithm 1 (see Algorithm 3).

---

**Algorithm 3** Improvement of Algorithm 1

---

**Input:** same as Algorithm 1

**Output:** same as Algorithm 1

```

1: for  $l = k + 1, k + 2, \dots, k + |W| + 2$  do
2:   if  $(P_l, P_{l-k})$  is clockwise hole-disjoint then
3:     return  $P_{l-k}, \dots, P_{l-1}, P_l$ , which are pairwise hole-disjoint paths
4:   else
5:     let  $P_{l+1}$  be the  $s$ - $t$  path in  $R(P_{l-k}, P_{l-k+1})$  such that  $(P_l, P_{l+1})$  is clockwise hole-disjoint and
        $R(P_{l+1}, P_{l-k+1})$  is maximized under this condition
6:     if  $P_{l'+1} = P_{l'-k+1}$  holds for  $l' = l - k + 1, \dots, l$  then
7:       break iterations and goto Line 11
8:     end if
9:   end if
10: end for
11: reports the non-existence of pairwise hole-disjoint  $k + 1$   $s$ - $t$  paths

```

---

**Theorem 3.** Algorithm 3, when used as a subroutine in Algorithm 2 in place of Algorithm 1, can find an optimal solution of the GMFCD.

*Proof.* It is sufficient to show that if there exists an integer  $l$  such that

- $(P_{l'}, P_{l'-k})$  is clockwise hole-disjoint, and
- $P_{l'+1} = P_{l'-k+1}$

for  $l' = l - k + 1, l - k + 3, \dots, l$  when Algorithm 1 is applied with  $k = \text{MAX-FLOW}$ , the above conditions also hold for  $l' = l + 1, l + 2, \dots, k + |W| + 2$ .

Consider the case of  $l' = l + 1$ . Since  $(P_{l-k+1}, P_{l-2k+1})$  is clockwise hole-disjoint,  $P_{l+1} = P_{l-k+1}$ , and  $P_{l-k+1} = P_{l-2k+1}$ ,  $(P_{l+1}, P_{l-k+1})$  is also clockwise hole-disjoint. Furthermore  $P_{l+2} = P_{l-k+2}$  also holds since  $P_{l-k+2} = P_{l-2k+2}$ ,  $P_{l-k+3} = P_{l-2k+3}$ ,  $\dots$ ,  $P_{l+1} = P_{l-k+1}$ .

By applying this argument repeatedly,  $(P_{l'}, P_{l'-k})$  is clockwise hole-disjoint and  $P_{l'+1} = P_{l'-k+1}$  for  $l' = l + 1, l + 2, \dots, k + |W| + 2$ .  $\square$

### 3.3 Convergence Property of the Improved Algorithm

The improvement that we gave in the previous subsection can reduce the running time of computing an optimal solution of the GMFCD only if  $P_{l+1} = P_{l-k+1}$  holds in some  $k$  consecutive iterations in Algorithm 3, where  $k = \text{MAX-FLOW}$ . This naturally raises the question whether there always exists an integer  $l$  such that  $P_{l'+1} = P_{l'-k+1}$  for  $l' = l - k + 1, \dots, l$  in Algorithm 3 with  $k = \text{MAX-FLOW}$ .

The objective of this subsection is to show that  $P_{l+1} = P_{l-k+2}$  holds for sufficiently large  $l$  under the assumption that  $\text{MIN-CUT} = \text{MAX-FLOW}$ . See Section 2 for the validity of this assumption. We mention that this question is for pure theoretical interest and that we can apply Algorithm 3 regardless of this issue.

**Theorem 4.** *Suppose that we are given a graph  $G = (V, E)$  drawn in the plane, two distinct nodes  $s$  and  $t$ , a hole radius  $r_B$ , and a protection radius  $r_P(> r_B)$ . Moreover suppose that  $\text{MIN-CUT} = \text{MAX-FLOW}$ . Then, even if Algorithm 3 is modified so as not to stop after the iteration where  $l = k + |W| + 2$  and when the modified algorithm is applied with  $k = \text{MAX-FLOW}$ , there exists an integer  $l$  such that  $P_{l'+1} = P_{l'-k+2}$  for  $l' = l - k + 1, \dots, l$ .*

*Proof.* Since  $\text{MIN-CUT} = \text{MAX-FLOW}(= k)$ , we can take a hole cut  $\{H_1, H_2, \dots, H_k\}$  of size  $k$ . By the definition of hole cuts, each of  $P_1, P_2, \dots, P_k$  intersects at least one hole of  $\{H_1, H_2, \dots, H_k\}$ . Since  $k$  paths  $P_1, P_2, \dots, P_k$  are pairwise hole-disjoint, no pairs of these paths intersect a single hole  $H_i$ . Therefore, each hole of  $H_1, H_2, \dots, H_k$  intersects one path of  $P_1, P_2, \dots, P_k$ , respectively. By changing the indices if necessary, suppose that path  $P_i$  intersects hole  $H_i$  for  $i = 1, 2, \dots, k$ . (see Figure 3).

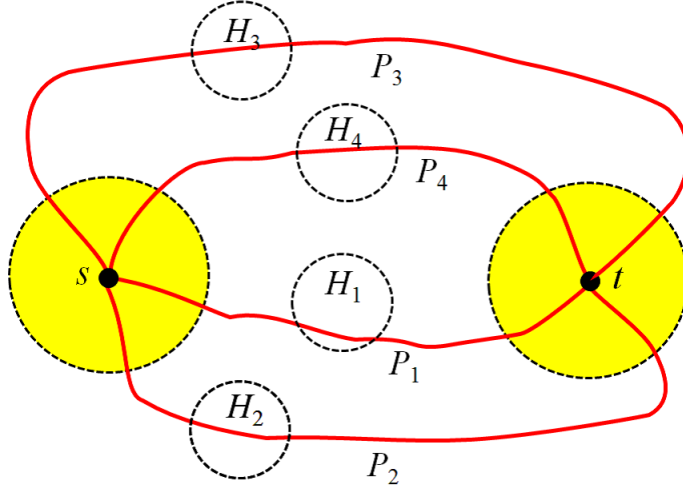


Figure 3: Correspondence between paths and holes.

For two  $s$ - $t$  paths  $P', P''$  that intersect a single hole  $H$ , we say “ $P'$  is on the left side of  $P''$  with respect to  $H$ ” if  $P'$  and  $P''$  do not cross with each other, and two nodes  $s, t$  are disconnected in  $R(P', P'') - H$ . Then, for  $i = 1, 2, \dots, k$  and  $j \geq 0$ , since path  $P_{i+(j+1)k}$  is in  $R(P_{i+jk-1}, P_{i+jk})$  and both  $P_{i+(j+1)k}$  and  $P_{i+jk}$  intersect hole  $H_i$ , path  $P_{i+(j+1)k}$  is on the left side of path  $P_{i+jk}$  with respect to hole  $H_i$  (see Figure 4).

For  $i = 1, 2, \dots, k$ , define

$$g_i(n) := \sum_{j=1}^{n-1} |R(P_{i+jk}, P_{i+(j-1)k})|.$$



## 4 Min-Cut Algorithms

### 4.1 Existing Min-Cut Algorithm

In this subsection we review the polynomial min-cut algorithm for the GMCCD given in [5]. The following theorem ensures that either  $\text{MIN-CUT} = \text{MAX-FLOW}$  or  $\text{MIN-CUT} = \text{MAX-FLOW} + 1$  holds, which plays a significant role in the min-cut algorithm.

**Theorem 5** ([5]). *Suppose that we are given a graph  $G = (V, E)$  drawn in the plane, two distinct nodes  $s$  and  $t$ , a hole radius  $r_B$ , and a protection radius  $r_P(> r_B)$ . Then,*

$$\text{MAX-FLOW} \leq \text{MIN-CUT} \leq \text{MAX-FLOW} + 1.$$

*Moreover, we can compute a hole-cut whose size is at most  $\text{MAX-FLOW} + 1$  in polynomial time.*

This theorem implies that in order to find an optimal solution of the GMCCD, it is sufficient to check whether there exists a hole-cut of size  $\text{MAX-FLOW}$  or not. In what follows, for  $i = 1, \dots, k$ , let  $\mathcal{F}_i$  be the set of all faces of  $G$  contained in  $R(P_{i-1}, P_i)$ , where  $P_0 := P_k$ .

**Theorem 6** ([5]). *An optimal solution of the GMCCD can be computed in polynomial time (by using Algorithm 4).*

---

#### Algorithm 4 Hole-disjoint min-cut algorithm

---

**Input:** pairwise hole-disjoint  $k$   $s$ - $t$  paths  $P_1, \dots, P_k$  and a hole cut of size  $k + 1$

**Output:** a hole cut of minimum size

- 1: construct the directed graph  $D = (\mathcal{F}, A)$  defined as follows:

$$\begin{aligned} \mathcal{F} &:= \bigcup_i \mathcal{F}_i, \\ A &:= \{(f_i, f_{i+1}) \mid i \in \{1, \dots, k\}, f_i \in \mathcal{F}_i, f_{i+1} \in \mathcal{F}_{i+1}, \\ &\quad \exists H \in \mathcal{H}(r_B, r_P) \text{ intersecting } f_i, P_i, \text{ and } f_{i+1}\}. \end{aligned}$$

- 2: **for all**  $f \in \mathcal{F}_1$  **do**
  - 3:   try to find a minimum-length directed cycle  $Q_f$  in  $D$  containing  $f$  by a breadth-first-search from  $f$
  - 4:   **if** the length of  $Q_f$  is  $k$  **then**
  - 5:     **return** a hole cut of size  $k$  corresponding to directed cycle  $Q_f$
  - 6:   **end if**
  - 7: **end for**
  - 8: **return** a hole cut of size  $k + 1$
- 

Note that if we have a directed cycle in  $D = (\mathcal{F}, A)$ , which is defined in Algorithm 4, we can obtain a hole cut whose size is the same as the length of the directed cycle. The running time of Algorithm 4 is bounded by  $O(|V||W|)$ , since the number of nodes and edges of the directed graph  $D = (\mathcal{F}, A)$  is  $|\mathcal{F}| = O(|V|)$  and  $|A| = O(|W|)$ , respectively. Thus, in Line 3 of Algorithm 4, finding a minimum-length directed cycle takes  $O(|\mathcal{F}_1|(|\mathcal{F}| + |A|)) = O(|V||W|)$  time. Practically, since  $|W| = O(|V|)$ , the running time of Algorithm 4 is  $O(|V|^2)$ .

## 4.2 Improvement of Min-Cut Algorithm

We improve the min-cut algorithm (Algorithm 4) of [5] for the GMCCD in this subsection. As we reviewed in the previous subsection, the running time of this min-cut algorithm is  $O(|V|^2)$  practically. The improvement does not change the theoretical computational time complexity, in the same way as our Improvement for the GMFCD. Practically, however, the running time of the algorithm is decreased drastically in most cases, as we will see in Section 5.2.

Let  $k := \text{MAX-FLOW}$ . First, we note that we can obtain pairwise hole-disjoint  $k$   $s$ - $t$  paths  $P_1, \dots, P_k$  satisfying the following property, by using the improved max-flow algorithm (see Section 3.3).

Property A: The first edges of  $P_1, \dots, P_k$ , which are incident to  $s$ , occur in this order clockwise at  $s$ , and for  $i = 1, \dots, k$ ,  $P_i$  is the  $s$ - $t$  path such that  $(P_{i-1}, P_i)$  is clockwise hole-disjoint and  $R(P_{i-1}, P_i)$  is minimized under this condition ( $P_0 := P_k$ ).

Here we use the following lemma, which we can show by applying the argument in the proof of Theorem 3 in [5] to the case with Property A.

**Lemma 7** ([5]). *Suppose that we are given a graph  $G = (V, E)$  drawn in the plane, two distinct nodes  $s$  and  $t$ , a hole radius  $r_B$ , and a protection radius  $r_P(> r_B)$ . Moreover suppose that there exist pairwise hole-disjoint  $k$   $s$ - $t$  paths  $P_1, \dots, P_k$  that satisfy Property A. Then, there exists a closed curve  $C$  which does not intersect with itself, separates  $s$  and  $t$ , and is a concatenation of  $J_1, L_1, J_2, L_2, \dots, J_k, L_k$  in this order, where*

- for  $j = 1, 2, \dots, k$ ,  $J_j$  is a curve contained in a face  $f_j \in \mathcal{F}_j$ , and face  $f_j$  and path  $P_j$  share an edge of  $G$  in common, and
- for  $j = 1, 2, \dots, k$ ,  $L_j$  is a line segment in  $\mathbb{R}^2$  that can be covered by a hole  $H \in \mathcal{H}(r_B, r_P)$ .

By using the lemma, we obtain Algorithm 5.

**Theorem 8.** *Suppose that we are given a graph  $G = (V, E)$  drawn in the plane, two distinct nodes  $s$  and  $t$ , a hole radius  $r_B$ , and a protection radius  $r_P(> r_B)$ . Moreover suppose that we have pairwise hole-disjoint  $k$   $s$ - $t$  paths  $P_1, \dots, P_k$  that satisfy Property A. Then, Algorithm 5 can find an optimal solution of the GMCCD.*

*Proof.* By Lemma 7, there exists a closed curve  $C$  which does not intersect with itself, separates  $s$  and  $t$ , and is a concatenation of  $J_1, L_1, J_2, L_2, \dots, J_k, L_k$  in this order, where

- for  $j = 1, 2, \dots, k$ ,  $J_j$  is a curve contained in a face  $f_j \in \mathcal{F}_j$ , and face  $f_j$  and path  $P_j$  share an edge of  $G$  in common, and
- for  $j = 1, 2, \dots, k$ ,  $L_j$  is a line segment in  $\mathbb{R}^2$  that can be covered by a hole  $H \in \mathcal{H}(r_B, r_P)$ .

Since each face  $f_j$  shares a common edge with path  $P_j$ , the directed graph  $D'$  in Algorithm 5, which is a subgraph of  $D$ , has a directed cycle of length  $k$ . Then, if a minimum-length directed cycle is found in  $D'$ , its length is  $k$  because  $D'$  has no directed cycle whose length is less than  $k$ . We can find a minimum-length directed cycle of  $D'$  by Algorithm 5, which completes the proof.  $\square$

Although the improvement does not change the theoretical computational time complexity, practically the running time is decreased drastically in most cases. We note that in order to use Algorithm 5, it is necessary that  $\text{MIN-CUT} = \text{MAX-FLOW}$  ( $= k$ ) and we have pairwise hole-disjoint  $k$   $s$ - $t$  paths  $P_1, \dots, P_k$  satisfying Property A. In most cases,  $\text{MIN-CUT} = \text{MAX-FLOW}$  holds and we can find such  $s$ - $t$  paths efficiently by using Algorithm 3.

---

**Algorithm 5** Improved hole-disjoint min-cut algorithm

---

**Input:** pairwise hole-disjoint  $k$   $s$ - $t$  paths  $P_1, \dots, P_k$  which satisfy Property A

**Output:** a hole cut of minimum size

1: construct the directed graph  $D' = (\mathcal{F}', A')$  defined as follows:

$$\begin{aligned}\mathcal{F}'_i &:= \text{the set of all faces in } \mathcal{F}_i \\ &\quad \text{that share a common edge of } G \text{ with } P_i, \\ \mathcal{F}' &:= \bigcup_i \mathcal{F}'_i, \\ A' &:= \{(f_i, f_{i+1}) \mid i \in \{1, \dots, k\}, f_i \in \mathcal{F}'_i, f_{i+1} \in \mathcal{F}'_{i+1}, \\ &\quad \exists H \in \mathcal{H}(r_B, r_P) \text{ intersecting } f_i, P_i, \text{ and } f_{i+1}\}.\end{aligned}$$

2: **for all**  $f \in \mathcal{F}'_1$  **do**

3:   try to find a minimum-length directed cycle  $Q_f$  in  $D'$  containing  $f$  by a breadth-first-search from  $f$

4:   **if** the length of  $Q_f$  is  $k$  **then**

5:     **return** a hole cut of size  $k$  corresponding to directed cycle  $Q_f$

6:   **end if**

7: **end for**

---

## 5 Experimental Results

In this section, we describe our experimental results. We implemented the algorithms for the GMFCD and the GMCCD, and evaluated their performance by computational experiments. Our experiments were conducted on a computer with Intel Core i7, 2.8 GHz and 8 GB of memory. All programs are implemented in Java.

### 5.1 Experimental Evaluation of the Algorithms

We evaluate the performance of the improved max-flow algorithm and the improved min-cut algorithm by comparing them with the existing algorithms.

We generated 10 input plane graphs with 100, 300, 1000, 3000, 10000, 30000, 100000 nodes, respectively, randomly in a  $300 \times 400$  rectangular. We randomly chose the positions of  $s$  and  $t$  in the rectangular and let  $r_B = 7$  and  $r_P = 30$ .

Figure 5 shows the result comparing the average running time of the improved max-flow algorithm with that of the existing max-flow algorithm to the 10 input graphs consisting of the respective number of nodes. Figure 6 shows the result comparing the average running time of the improved min-cut algorithm with that of the existing min-cut algorithm, in the same way. We can see that the running time of the algorithms were improved from  $O(|V|^{2.172})$  to  $O(|V|^{1.266})$  for the GMFCD, and from  $O(|V|^{1.811})$  to  $O(|V|^{1.162})$  for the GMCCD, where we estimated the time complexity by the least-squares method (in double-logarithmic scale).

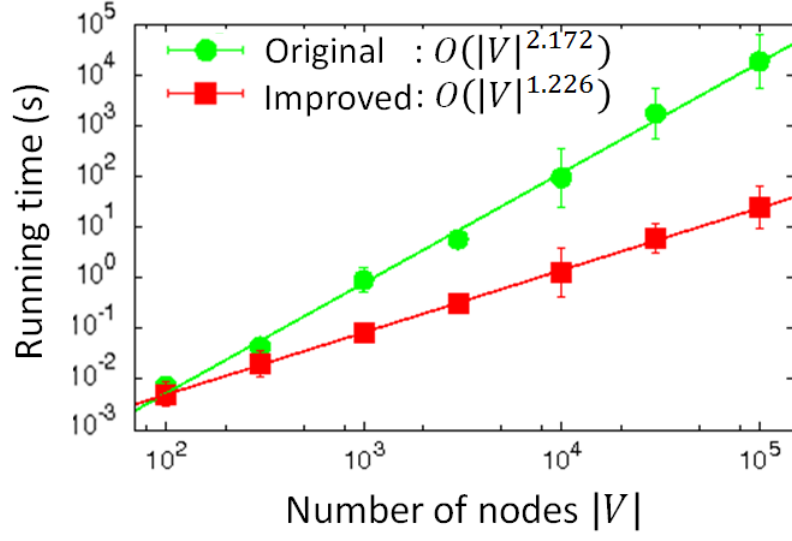


Figure 5: Relationship between the running time for the GMFCD and the numbers of nodes

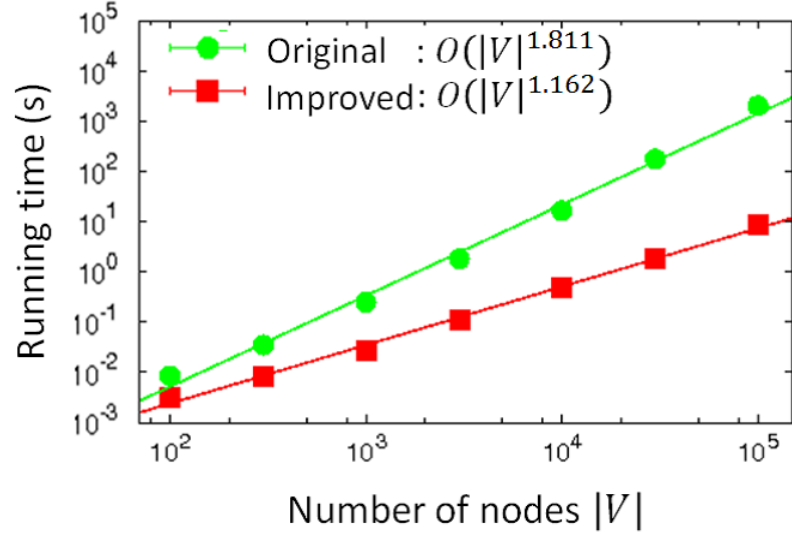


Figure 6: Relationship between the running time for the GMCCD and the numbers of nodes

## 5.2 Application to NewYork Road Network

As we can see in the previous subsection, our improvements of the algorithms for the GMFCD and the GMCCD enable us to find in a realistic time optimal solutions in huge networks with more than 100,000 nodes. As a practical instance, we analyze reliability between John F. Kennedy International Airport (JFK) and Newark Liberty International Airport (EWR) in NewYork road network [13] consisting of 264,346 nodes and 366,923 edges.

We set two nodes  $s$  and  $t$  to be JFK and EWR, respectively, and let the hole radius  $r_B$  be 2km (approximately), and the protection radius  $r_P$  be about 7km (approximately). The solutions of the GMFCD and the GMCCD are shown in Figure 7. We can see  $\text{MAX-FLOW} = \text{MIN-CUT} = 4$ . It took 32 seconds to solve the GMFCD by Algorithm 3, and 14 seconds to solve the GMCCD by Algorithm 5 (after solving the GMFCD).

The solutions show that even if three failures of radius 2km happen simultaneously, at least one transit routes remain intact. We can also see that all the disks of the optimal hole cut are placed on the bridges, which means that the bridges are vulnerable when we consider reliability of NewYork road network between JFK and EWR.

Here we focus on vulnerability of bridges. We can expect that if we protect one of the bridges, that is, if we introduce a new protective disk around the bridge and prohibit placing a hole whose center is contained in the protective disk, connectivity of the network will increase. Indeed, Figure 8 shows that connectivity increases from 4 to 5 by protecting the bridge.

In this way, by using our algorithms, we can not only find optimal solutions of the GMFCD and the GMCCD, but also identify vulnerable regions we should protect particularly.

We also note that our algorithms can be applied to the case with new protective disks, or with holes of different shapes under some conditions. More specifically, we can replace  $\mathcal{H}(r_B, r_P)$  with any set  $\mathcal{H}$  of convex shapes satisfying the property below (see the concluding remarks in [5]).

Property B: If two line segments  $L_1$  in  $H_1 \in \mathcal{H}$  and  $L_2$  in  $H_2 \in \mathcal{H}$  are crossing, then,  $H_1$  contains an endpoint of  $L_2$  or  $H_2$  contains an endpoint of  $L_1$ .

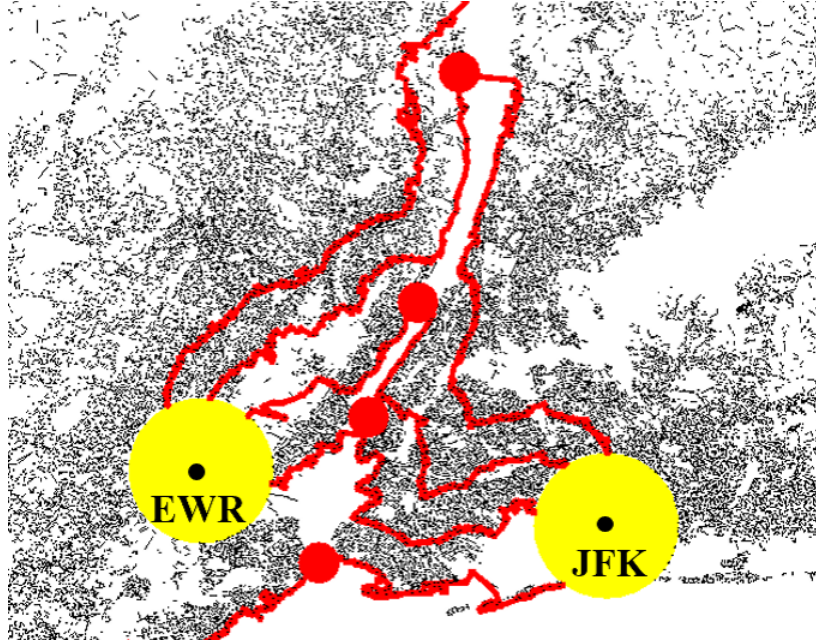


Figure 7: Case that the bridges are not protected:  $\text{MIN-CUT} = 4$ . Note that the figure shows a part of NewYork road network. The path starting from the lower-left part of the protective disk around JFK goes down outside the figure once, travels half round NewYork, comes in from the top of the figure, and finally reaches EWR.



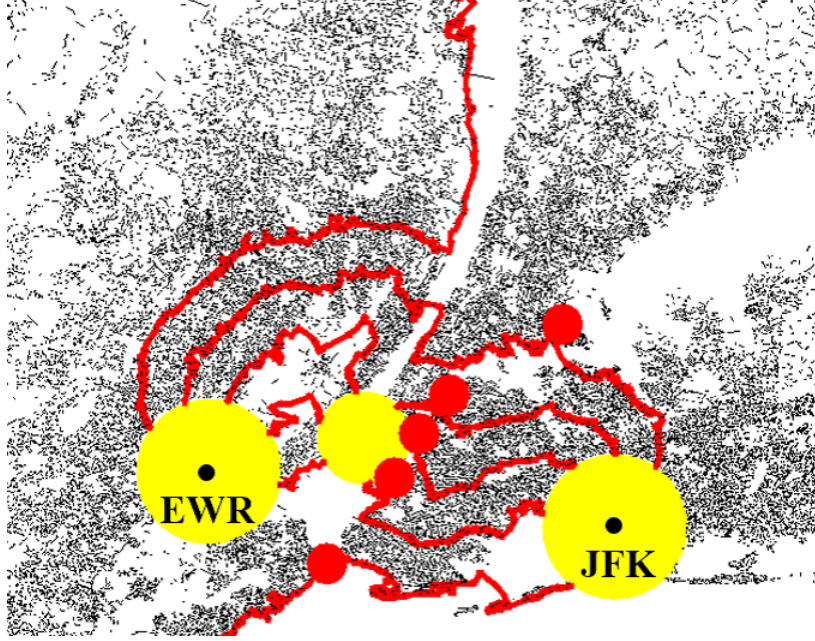


Figure 8: Case that a bridge is protected: MIN-CUT = 5.

### 5.3 Shortening Hole-disjoint Paths

In the previous subsection, we successfully found optimal solutions of the GMFCD and the GMCCD in NewYork road network. However, each path of the hole-disjoint flow looks quite unnatural. In particular, as we can see in Figures 7 and 8, the path starting from the lower-left part of the protective disk around JFK goes down outside the figure once, travels half round NewYork, comes in from the top of the figure, and finally reaches EWR, which is extremely long. Since a solution of the GMFCD represents a set of transit routes, it is not desirable that each transit route is unnecessarily long.

In this subsection, we shorten the paths by the following local search method. Let  $k := \text{MAX-FLOW}$ . We choose a path  $P$  from among pairwise hole-disjoint  $k$   $s$ - $t$  paths  $P_1, P_2, \dots, P_k$ , fix the remaining  $k - 1$  paths, and then find a shortest path  $P'$  which is hole-disjoint from any of the remaining  $k - 1$  paths by a depth-first-search, and replace path  $P$  with another path  $P'$ , repeatedly, until the set of the paths is not updated.

We note that this algorithm stops in finite iterations since the total number of edges contained in  $P_1, P_2, \dots, P_k$  is non-increasing. Figure 9 shows the result of this algorithm applied to Figure 7. We can see that each path turns shorter. The number of iterations was 12 in this case.

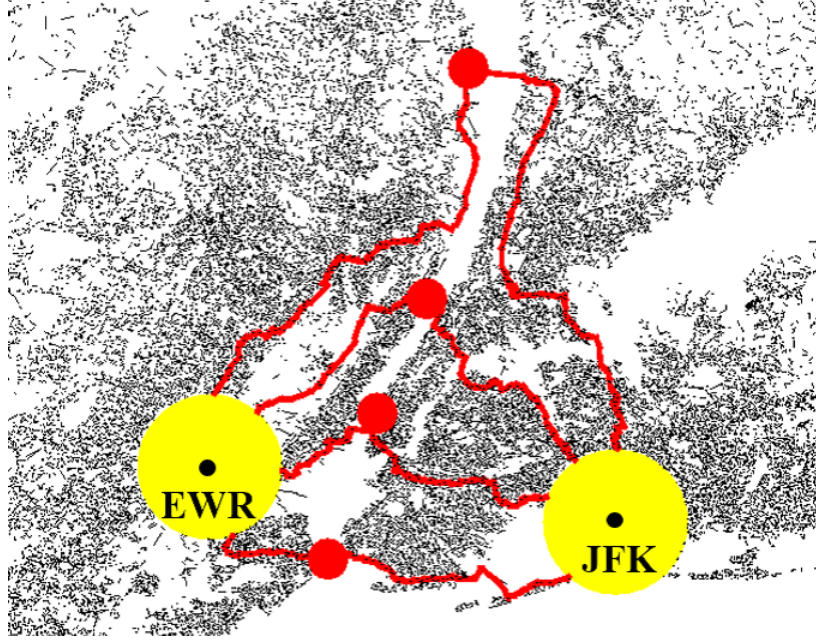


Figure 9: Result that we shorten the hole-disjoint paths in Figure 7.

## 6 Conclusion

In this paper, we discussed the geographical max-flow and the geographical min-cut in the Circular Disk Failure Model introduced by Neumayer–Efrat–Modiano [7]. We improved the algorithms to find optimal solutions of the GMFCD and the GMCCD given by Kobayashi–Otsuki [5]. We also evaluated the performance of these algorithms by computational experiments. The running time of the algorithms was improved from  $O(|V|^{2.172})$  to  $O(|V|^{1.266})$  for the GMFCD, and from  $O(|V|^{1.811})$  to  $O(|V|^{1.162})$  for the GMCCD in random instances.

The improvements enable us to compute in a realistic time connectivity of huge networks consisting of more than 100,000 nodes. We analyzed connectivity of New York road network with 264,346 nodes. We showed that by using our algorithms, we can not only find optimal solutions of the GMFCD and the GMCCD, but also identify vulnerable regions we should protect particularly. Moreover, we proposed a heuristic based on a local search method to shorten each path of a hole-disjoint flow.

## References

- [1] P. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, “The resilience of WDM networks to probabilistic geographical failures,” *IEEE/ACM Transactions on Networking*, vol. 21 (2013), pp. 1525–1538.
- [2] D. Bienstock, “Some generalized max-flow min-cut problems in the plane,” *Mathematics of Operations Research*, vol. 16 (1991), pp. 310–333.
- [3] L. R. Ford and D. R. Fulkerson, “Maximal flow through a network,” *Canadian Journal of Mathematics*, vol. 8 (1956), pp. 399–404.

- [4] M. F. Habib, M. Tornatore, F. Dikbiyik, and B. Mukherjee, “Disaster survivability in optical communication networks,” *Computer Communications*, vol. 36 (2013), pp. 630–644.
- [5] Y. Kobayashi and K. Otsuki, “Max-flow min-cut theorem and faster algorithms in a circular disk failure model,” in *Proc. IEEE INFOCOM*, 2014, pp. 1635–1643.
- [6] K. Menger, “Zur allgemeinen Kurventheorie,” *Fundamenta Mathematicae*, vol. 10 (1927), pp. 96–115.
- [7] S. Neumayer, A. Efrat, and E. Modiano, “Geographic max-flow and min-cut under a circular disk failure model,” in *Proc. IEEE INFOCOM*, 2012, pp. 2736–2740.
- [8] S. Neumayer and E. Modiano, “Network reliability with geographically correlated failures,” in *Proc. IEEE INFOCOM*, 2010, pp. 1658–1666.
- [9] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, “Assessing the vulnerability of the fiber infrastructure to disasters,” *IEEE/ACM Transactions Networking*, vol. 19 (2011), pp. 1610–1623.
- [10] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Algorithms and Combinatorics, vol. 24, Springer, Berlin, 2003.
- [11] A. Sen, S. Murthy, and S. Banerjee, “Region-based connectivity — a new paradigm for design of fault-tolerant networks,” in *Proc. HPSR*, 2009, pp. 94–100.
- [12] S. Trajanovski, F. A. Kuipers, and P. V. Mieghem, “Finding critical regions in a network,” in *Proc. NetSciCom*, 2013, pp. 223–228.
- [13] 9th DIMACS Implementation Challenge — Shortest Paths,  
<http://www.dis.uniroma1.it/challenge9/download.shtml>