MATHEMATICAL ENGINEERING TECHNICAL REPORTS

Max-Flow Min-Cut Theorem and Faster Algorithms in a Circular Disk Failure Model

Yusuke KOBAYASHI and Kensuke OTSUKI

METR 2013-15 August 2013

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

Max-Flow Min-Cut Theorem and Faster Algorithms in a Circular Disk Failure Model*

Yusuke KOBAYASHI† kensuke OTSUKI

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

August 2013

Abstract

Fault-tolerance is one of the most important facors in designing networks. Failures in networks are sometimes caused by an event occuring in specific geographical regions such as hurricanes, earthquakes, bomb attacks, and Electromagnetic Pulse (EMP) attacks. In INFOCOM 2012, Neumayer et al. introduced geographical variants of max-flow min-cut problems in a circular disk failure model, in which each failure is represented by a disk with a predetermined size. In this paper, we solve two open problems in this model: we give a first polynomial-time algorithm for the geographic max-flow problem, and prove a conjecture of Neumayer et al. on a relationship between the geographic max-flow and the geographic min-cut.

1 Introduction

Fault-tolerance is one of the most important facors in designing networks. In most studies on fault-tolerance in networks, "connectivity" of the network is regarded as the measure of robustness (e.g. [1, 3, 4]). However, failures in networks are sometimes caused by an event occuring in specific geographical regions such as hurricanes, earthquakes, bomb attacks, and Electromagnetic Pulse (EMP) attacks. Recently, some models in which such localized failures are taken into consideration are proposed in [7, 8, 9, 11, 10].

In particular, Neumayer et al. [7] considered the model in which each failure is represented by a hole (disk) with a predetermined size (see Section 2 for details), and they gave a polynomial-time algorithm for computing the minimum number of failures that disconnect two specified nodes s and t, which they call the "geographic min-cut". They also formulated the problem of finding the maximum number of s-t paths such that no two paths can be disconnected by the same hole, which they call the "geographic max-flow". Similar problems are also considered in [2].

In this paper, we further develop theory and algorithms for geographic min-cut and geographic max-flow in this model. Our contributions are described as follows.

*This work was partially supported by JST, ERATO, Kawarabayashi Large Graph Project, Japan.
†Supported by Grant-in-Aid for Scientific Research, MEXT, Japan.

- We give a min-max theorem that characterizes a geographic max-flow (Theorem 1).
- We show that the geographic min-cut is at most the geographic max-flow plus one (Theorem 3), which was conjectured by Neumayer et al.
- We give a first polynomial-time algorithm for the geographic max-flow (Theorem 4).
- We give a polynomial-time algorithm for the geographic min-cut which is simpler (and probably faster) than known algorithms (Theorem 5).
- We implement our algorithms and confirm that they can solve large problems efficiently.

We emphasize here that the second and the third results solve two open problems raised by Neumayer et al. [7] in INFOCOM 2012.

We also note that we can extend our results to the case when each hole is of different shapes. See Section 6 for details.

This paper is organized as follows. First, we describe fromal problem settings in Section 2. Next, in Section 3, we discuss min-max relations between the geographic min-cut and the geographic max-flow, and prove the conjecture of Neumayer et al. In Section 4, we give algorithms for the geographic max-flow and the geographic min-cut based on the min-max relation. Then, experimental results are shown in Section 5. Finally, we give concluding remarks in Section 6.

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 $s, t \in V$, where each link is drawn as a line segment. In what follows, a link is sometimes called an edge. Let r_b be a hole radius and $r_p(> r_b)$ be a protection radius. A disk of radius r_p whose center is s or t is called a *protective disk*. Define $\mathcal{H}(r_b, r_p)$ as the set of all disks of radius r_b whose centers are not contained in protective disks of radius r_p. Each element of $\mathcal{H}(r_b, r_p)$ is called a *hole* in this paper.

We consider a geographic variant of the min-cut problem, which is defined as follows.

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.

Let MIN-CUT denote the optimal value of this problem. A set of holes in $\mathcal{H}(r_b, r_p)$ that disconnect s from t is called a *hole cut* in this paper. We can also consider a geographic variant of the max-flow problem,

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.

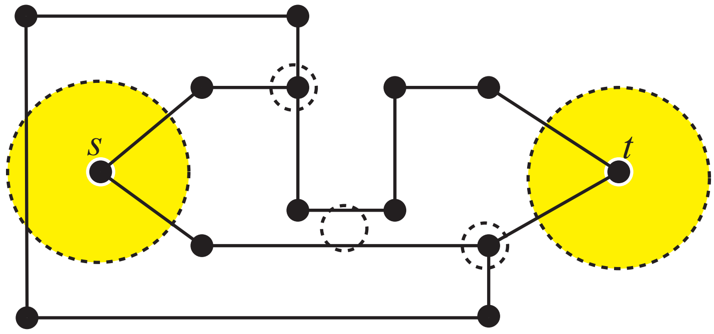

Figure 1: Example of the problems

Let MAX-FLOW denote the optimal value of this problem. So far, no polynomial-time algorithm for GMFCD was known, whereas a polynomial-time algorithm for GMCCD was given in [7].

Example 1. Consider the graph as in Fig. 1 (which is an example given in [7]). Small circles represent holes of radius r_b and two large shaded (or yellow in the color version) circles are protective disks. In the graph, we can easily see that MAX-FLOW = 1 and MIN-CUT = 2.

3 Geographic Max-Flow Min-Cut Theorem

In this section, we investigate relations between MAX-FLOW and MIN-CUT. First we give a characterization of maximum flows in Sections 3.1 and 3.2. Then, in Section 3.3, we prove MIN-CUT $\leq$ MAX-FLOW + 1, which is the conjecture of Neumayer et al. Note that this bound is tight by Example 1, and it significantly improves previously known bound: MIN-CUT $\leq 2 \cdot$ MAX-FLOW + 2 given in [2].

3.1 Statement of the theorem

Let C be a closed curve in the plane that does not go through s or t. We define the *winding number* $w(C)$ of C as the number of times that C separates s and t. More precisely, let L be the line segment connecting s and t, and fix orientations of L and C. Let $w_1(C)$ be the number of times that C crosses L from left to right and $w_2(C)$ be the number of times that C crosses L from right to left. Then, define $w(C) := |w_1(C) - w_2(C)|$.

We say that a closed curve C can be represented as an *alternating curve of length* l if it is a concatenation of $J_1, L_1, J_2, L_2, \dots, J_l, L_l$ in this order such that

- $J_1, J_2, \dots, J_l$ are curves each contained in a face of G, and
- For each $i = 1, 2, \dots, l$, L_i is a line segment that can be covered by a hole in $\mathcal{H}(r_b, r_p)$.

Note that J_i and L_i might be a single point. For a closed curve C, let $l(C)$ be the minimum number l such that C can be represented as an alternating curve of length l.

By using these notations, we can give an exact min-max theorem for MAX-FLOW, whose proof is given in the next subsection.

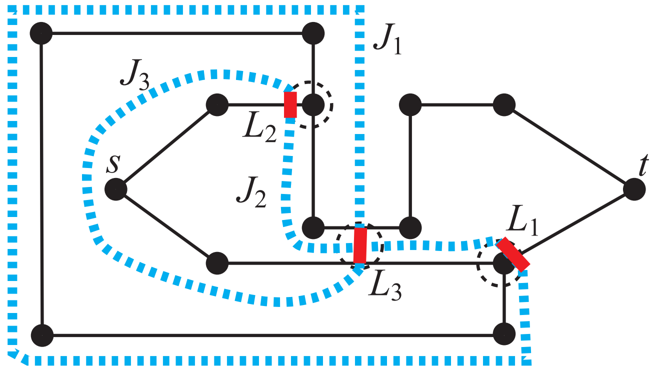

Figure 2: Example of a closed curve C

Theorem 1. *Suppose 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,*

$$\text{MAX-FLOW} = \min \left\{ \left\lfloor \frac{l(C)}{w(C)} \right\rfloor \mid C \textit{ is a closed curve} \right\}.$$

Example 2. Consider again the graph as in Fig. 1. As shown in Fig. 2, there exists a closed curve C such that $l(C) = 3$, $w(C) = 2$, and hence $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = 1$. This value is equal to MAX-FLOW.

3.2 Proof of Theorem 1

In this subsection, we give a proof of Theorem 1. Our proof is based on ideas in [6] (see also [5]), which shows a min-max theorem for maximum induced disjoint s-t paths in plane graphs.

We say that two s-t paths are *separated*, if no hole in $\mathcal{H}(r_b, r_p)$ intersects both of these paths. For a pair of edges $e, e' \in E$, if there exists a hole in $\mathcal{H}(r_b, r_p)$ that intersects both edges, then we take two points $w_{e,e'}$ on e and $w_{e',e}$ on e' that are contained in the common hole. Define

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

Let $\mathcal{L}$ be the set of all line segments with both endpoints in W such that each line segment is contained in a hole in $\mathcal{H}(r_b, r_p)$. Note that a line segment might be a single point, that is, $(w, w) \in \mathcal{L}$ for $w \in W$. Then, two s-t paths P and P' are not separated if and only if there exists a line segment $(w, w') \in \mathcal{L}$ such that w is on P and w' is on P'. Theoretically, $|W|$ is bounded by $|E|^2$ which is a polynomial size. Practially, we can obtain W by adding a small number of points to V, because we have to add points to V only in some exceptional cases (see e.g. Fig. 3).

First, we show

$$\text{MAX-FLOW} \leq \min \left\{ \left\lfloor \frac{l(C)}{w(C)} \right\rfloor \mid C \text{ is a closed curve} \right\}.$$

Suppose we have s-t paths $P_1, \dots, P_k$ that are pairwise separated and let C be a closed curve that is a concatenation of $J_1, L_1, J_2, L_2, \dots, J_{l(C)}, L_{l(C)}$ in this order. Since C intersects each P_i at least $w(C)$ times, each P_i intersects at least $w(C)$ line segments of $L_1, L_2, \dots, L_{l(C)}$. This means that $l(C) \geq k \cdot w(C)$. By the integrality of k, we have $k \leq \left\lfloor \frac{l(C)}{w(C)} \right\rfloor$.

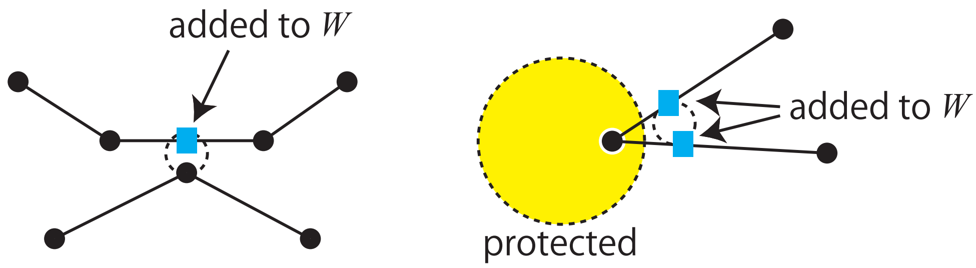

Figure 3: Construction of W

Next we show

$$\text{MAX-FLOW} \geq \min \left\{ \left\lfloor \frac{l(C)}{w(C)} \right\rfloor \mid C \text{ is a closed curve} \right\}$$

by giving an algorithm for finding either k pairwise separated s-t paths or a closed curve C with $\frac{l(C)}{w(C)} < k$ for any k. For any s-t path P, we suppose that it is oriented from s to t. For two s-t paths P' and P'' without crossings, let $R(P', P'')$ denote the closed region encircled by the closed curve $P' \cdot (P'')^{-1}$ in clockwise orientation. For two s-t paths P' and P'' without crossings, a pair (P', P'') is *clockwise separated* if for any hole H in $\mathcal{H}(r_b, r_p)$, $R(P', P'') - H$ is connected. Obviously, a pair (P', P'') is clockwise separated if P' and P'' are separated. In what follows, we show the inequality by the induction on k.

3.2.1 Induction step

First, we consider the case $k \geq 3$ under the assumption that we have $k-1$ pairwise separated s-t paths $P_1, \dots, P_{k-1}$. We may assume that these paths do not cross each other, and the first edges of $P_1, \dots, P_{k-1}$ occur in this order clockwise at s. Let P_k be an s-t path in $R(P_{k-1}, P_1)$ such that (P_{k-1}, P_k) is clockwise separated[1]. In our algorithm, we start with the k paths $P_1, \dots, P_{k-1}, P_k$, and we replace one path with a new path, repeatedly. Our algorithm is described in Algorithm 1 (see Fig. 4 for an example).

Algorithm 1 Induction step

Input: pairwise separated $k-1$ s-t paths $P_1, \dots, P_{k-1}$ and a path P_k defined as above
Output: pairwise separated k s-t paths or a closed curve C with $\frac{l(C)}{w(C)} < k$

```
for l = k, k+1, ..., k+|W|+1 do
    if (P_l, P_{l-k+1}) is clockwise separated¹ then
        return P_{l-k+1}, ..., P_{l-1}, P_l that are separated paths
    else
        let P_{l+1} be the s-t path in R(P_{l-k+1}, P_{l-k+2}) such that (P_l, P_{l+1}) is clockwise separated and
            R(P_{l+1}, P_{l-k+2}) is maximized under this condition
    end if
end for
return a closed curve C with l(C)/w(C) < k
```

---

[1]This condition is eqivalent to "separated" when $k \geq 3$. Since we will use the same argument for the case of $k = 2$ later, we use the condition "clockwise separated" here.

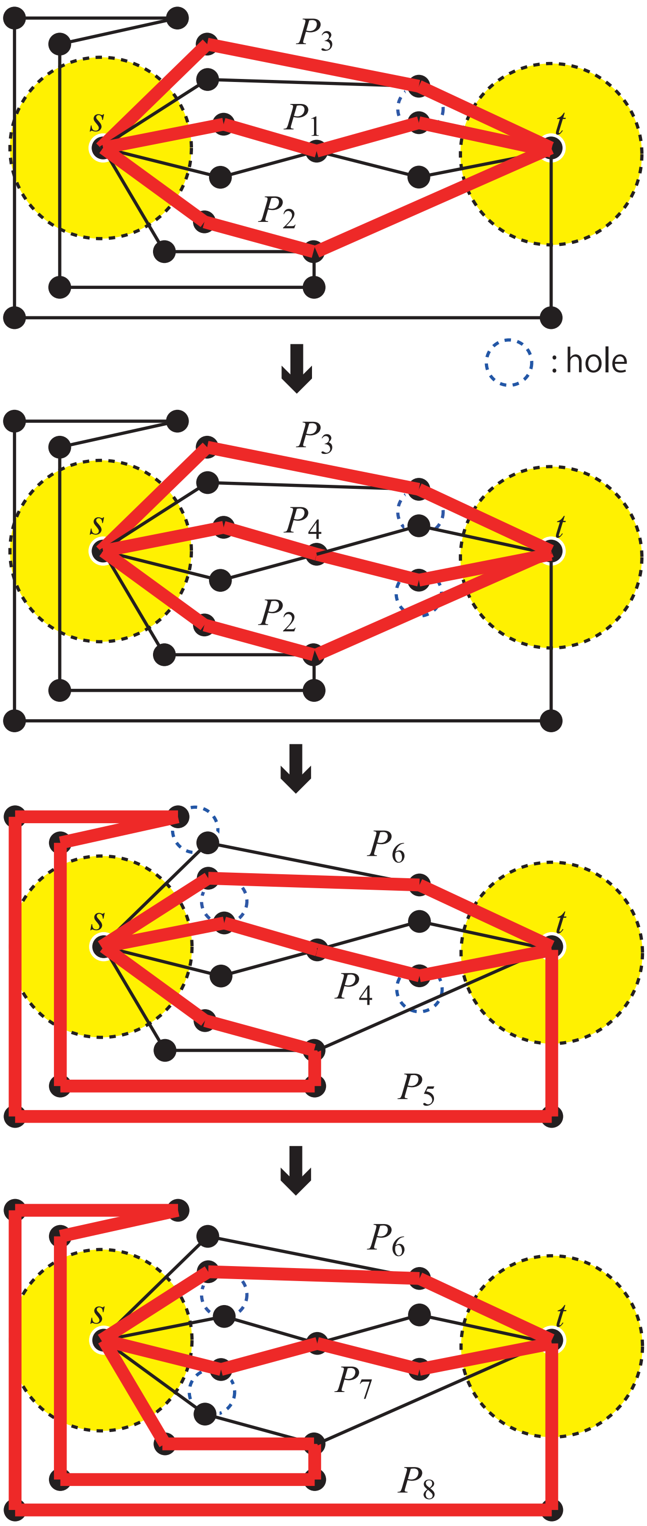

Figure 4: Iterations in Algorithm 1 ($k = 3$)

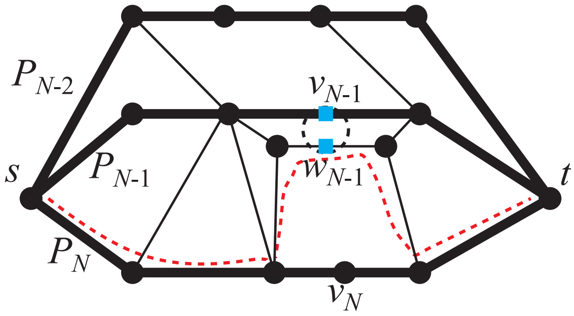

Figure 5: Definition of w_{N-1} and v_{N-1}

If we find pairwise separeted paths $P_{l-k+1}, \dots, P_{l-1}, P_l$ in line 3 of Algorithm 1, then we are done. In what follows, we give a procedure for finding a closed curve C with $\frac{l(C)}{w(C)} < k$ (line 8) when such paths do not appear while $l = k, k+1, \dots, k+|W|+1$.

Let $N := k + |W| + 1$. By the assumption, (P_N, P_{N-k+1}) is not clockwise separated, and hence there exists a node $v_N \in P_N \setminus P_{N-k}$. Note that a path is regarded as a subset of the plane. Since P_N maximizes $R(P_N, P_{N-k+1})$, if we reroute P_N so that the obtained path does not go through v_N, then the path contains a node close to P_{N-1}. More precisely, as in Fig. 5, we can find a pair of nodes $w_{N-1} \in W$ and $v_{N-1} \in P_{N-1} \cap W$ such that $(w_{N-1}, v_{N-1}) \in \mathcal{L}$ (i.e., w_{N-1} and v_{N-1} are covered by a common hole in $\mathcal{H}(r_b, r_p)$) and v_N and w_{N-1} can be connected by a curve J_N contained in a face of G. Furthermore, we can see that $v_{N-1} \notin P_{N-k-1}$, because w_{N-1} is strictly to the right of P_{N-k} (when we walk from s to t along P_{N-k}). By repeating the same argument, we can find v_i, w_i, and J_i for $i = N-1, N-2, \dots, k+1$ such that

- $w_i \in W$ and $v_i \in (P_i \setminus P_{i-k}) \cap W$ with $(w_i, v_i) \in \mathcal{L}$, and
- J_i is a curve from v_i to w_{i-1} contained in a face of G.

By pigeonhole principle, $v_i = v_j$ for some $k+1 \leq i < j \leq N$. Let C be a closed curve obtained by concatenating

$$(v_i, w_i), J_{i+1}, (v_{i+1}, w_{i+1}), J_{i+2}, \dots, (v_{j-1}, w_{j-1}), J_j$$

in this order, where (x, y) is the line segment connecting x and y. We will show that this curve C satisfies $\frac{l(C)}{w(C)} < k$, which is equivalent to $u := \left\lfloor \frac{j-i}{k} \right\rfloor < w(C)$, because $l(C) = j - i$. If $u = 0$, then the inequality is trivial. Otherwise, v_j is strictly to the right of P_{j-k}. When we consider a curve from v_i on P_i to v_{j-k} on P_{j-k} along C, it separates s and t at least $u - 1$ times, because $j - k \geq i + (u-1)k$. Therefore, C separates s and t more than u times, that is, $w(C) > u$.

By the above procedure, we can find a closed curve C with $\frac{l(C)}{w(C)} < k$ in line 8 of Algorithm 1.

3.2.2 Base cases

Next we deal with the base cases ($k = 1, 2$) of the induction. Since the case of $k = 1$ is trivial, we consider the case when $k = 2$. We show the following claim.

Claim 2. *Suppose that s and t are connected in G. We can find in polynomial time either*

- *a hole in $\mathcal{H}(r_b, r_p)$ that disconnects s and t, or*

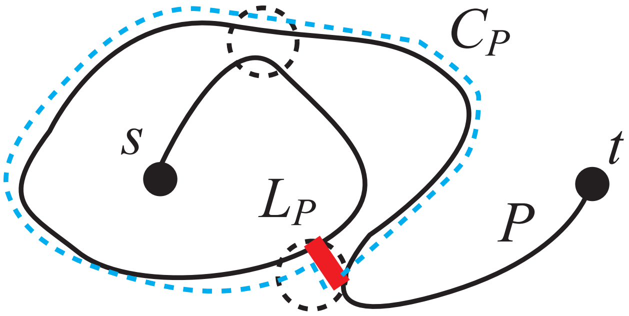

Figure 6: Definition of L_Q and C_Q

- *an s-t path P in G such that for any line segment $L \in \mathcal{L}$, the union of P and L (which is regarded as a subset of the plane) contains no closed curve separating s and t.*

Proof. Assume that G contains no s-t path satisfying the condition in the second case. That is, for any s-t path P, there exists a line segment $L \in \mathcal{L}$ such that $P \cup L$ contains a closed curve separating s and t.

For each s-t path P, we take a line segment $L_P \in \mathcal{L}$ and a closed curve C_P in $P \cup L_P$ such that the enclosed region containing s, which we denote $\mathrm{ins}(C_P)$, is as large as possible (see Fig. 6). Among all s-t paths, we choose an s-t path P^* such that $\mathrm{ins}(C_{P^*})$ is minimal.

Now we find a path from s to $P^* \cap C_{P^*}$. If such a path Q exists, then by concatenating Q and a subpath of P^*, we obtain an s-t path P' such that $\mathrm{ins}(C_{P'})$ is strictly contained in $\mathrm{ins}(C_{P^*})$, which contradicts the minimality of $\mathrm{ins}(C_{P^*})$. Therefore, there exists no path from s to $P^* \cap C_{P^*}$, which means that the hole containing L_{P^*} satisfies the condition in the first case.

Note that this argument also gives an algorithm. We begin with an arbitrarily s-t path P_0, and find an s-t path P' with smaller $\mathrm{ins}(C_{P'})$. By repeating this procedure, we can find a hole or an s-t path P satisfying the condition. □

If we have a hole in $\mathcal{H}(r_b, r_p)$ that disconnects s and t, then we have a closed curve C with $l(C) = w(C) = 1$. Otherwise, we have an s-t path P in G as in the second case of Claim 2. In this case, define $P_1 = P_2 = P$ and assume that P_2 is to the left of P_1. Then, since (P_1, P_2) is clockwise-separated, we can apply Algorithm 1 to obtain pairwise separated two s-t paths or a closed curve C with $\frac{l(C)}{w(C)} < 2$, which completes the proof for the case of $k = 2$.

3.3 Proof of the conjecture

By using Theorem 1, in this subsection we give a proof of the conjecture of Neumayer et al.

Theorem 3. *Suppose 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,*

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

Proof. Since MAX-FLOW $\leq$ MIN-CUT is obvious, we prove MIN-CUT $\leq$ MAX-FLOW + 1. By Theorem 1, we can take a closed curve C such that $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor =$ MAX-FLOW. Hence, it suffices to find a hole cut of size $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor + 1$ (i.e., a set of $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor + 1$ holes in $\mathcal{H}(r_b, r_p)$ that disconnect s from t).

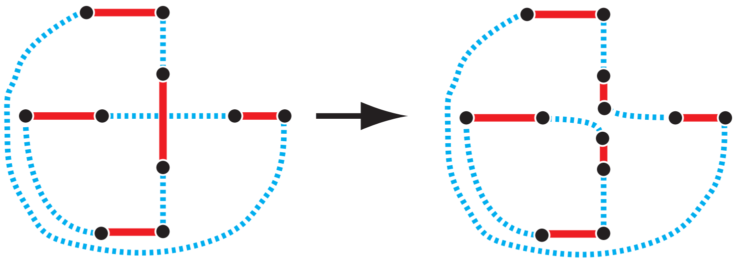

Figure 7: Uncrossing procedure 1

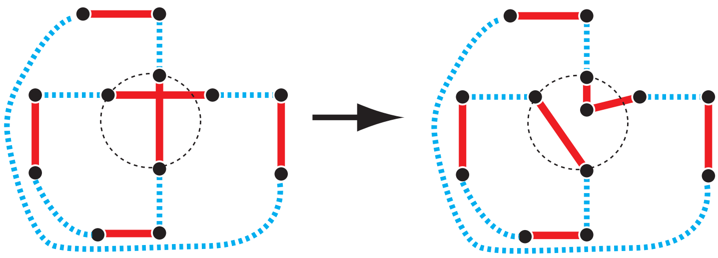

Figure 8: Uncrossing procedure 2

If $w(C) \geq 2$, then C must contain a self-crossing and we can decompose C into two closed curves C_1 and C_2 by uncrossing procedures (see Figures 7 and 8). Obviously, $w(C_1) + w(C_2) = w(C)$. To evaluate the total length $l(C_1) + l(C_2)$, we consider the following three cases.

1. If two curves J_i and J_j are corssing, then we can easily uncross C without increasing the length, that is, $l(C_1) + l(C_2) = l(C)$.

2. If a curve J_i and a line segment L_j are corssing, then we can uncross C by using two line segments instead of L_j, that is, $l(C_1) + l(C_2) \leq l(C) + 1$ (see Fig. 7).

3. Suppose that two line segments L_i and L_j are corssing. Then, the hole containing L_i also conatins an endnode of L_j or the hole containing L_j also conatins an endnode of L_i. Therefore, we can uncross C by using at most three line segments instead of L_i and L_j, that is, $l(C_1) + l(C_2) \leq l(C) + 1$ (see Fig. 8).

In each case, we have $l(C_1) + l(C_2) \leq l(C) + 1$. By repeating uncrossing procedures, we have closed curves $C_1, C_2, \ldots, C_{w(C)}$ such that $w(C_i) = 1$ for each i and $\sum_i l(C_i) \leq l(C) + w(C)$. Since we have

$$\min_i \{l(C_i)\} \leq \left\lfloor \frac{1}{w(C)} \sum_i l(C_i) \right\rfloor \leq \left\lfloor \frac{l(C)}{w(C)} \right\rfloor + 1,$$

there exists a closed curve C_i such that $w(C_i) = 1$ and $l(C_i) \leq \left\lfloor \frac{l(C)}{w(C)} \right\rfloor + 1$. This shows the existence of a hole cut of size $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor + 1$. □

4 Algorithms

In this section, we discuss algorithmic results on the GMFCD and the GMCCD. First, by the constructive proof of Theorem 1 in Section 3.2, we obtain a polynomial-time algorithm for computing MAX-FLOW.

Theorem 4. *An optimal solution of the GMFCD and a closed curve C minimizing $\lfloor \frac{l(C)}{w(C)} \rfloor$ can be computed in polynomial time.*

Note that this is the first polynomial-time algorithm for the GMFCD. The most time consuming part is Algorithm 1 that runs in $O(|W|^2)$ time. Since we execute Algorithm 1 at most $k :=$ MAX-FLOW times, the total running time is $O(k|W|^2)$. In most practical cases, since k is small and $|W| = O(|V|)$, the running time is $O(|V|^2)$.

Next, we propose a new algorithm for the GMCCD, which is simpler (and probably faster) than known algorithms.

Theorem 5. *An optimal solution of the GMCCD can be computed in polynomial time.*

Proof. By Theorem 4, we can compute s-t paths $P_1, \ldots, P_k$ that are mutually separated, where $k :=$ MAX-FLOW. Furthermore, by Theorem 3, we can also obtain a hole cut of size $k+1$ (i.e., a set of $k+1$ holes in $\mathcal{H}(r_b, r_p)$ that disconnect s from t). Since MAX-FLOW $\leq$ MIN-CUT, our remaining task is to find a hole cut of size k if one exists.

Since the case of $k = 1$ is easy, in what follows we suppose $k \geq 2$. We may assume that $P_1, \ldots, P_k$ do not cross each other, and the first edges of $P_1, \ldots, P_k$ occur in this order clockwise at s. Recall that $R(P_{i-1}, P_i)$ is the closed region encircled by the closed curve $P_{i-1} \cdot (P_i)^{-1}$ in clockwise orientation. For $i = 1, \ldots, k$, let $\mathcal{F}_i$ be the set of all faces of G contained in $R(P_{i-1}, P_i)$, where $P_0 := P_k$.

We observe that a hole cut of size k exists if and only if there exists a closed curve C with $w(C) = 1$ and $l(C) = k$ that is represented as a concatenation of $J_1, L_1, J_2, L_2, \ldots, J_k, L_k$ in this order, where J_i is a curve contained in a face of $\mathcal{F}_i$, and $L_i \in \mathcal{L}$ is a line segment connecting $R(P_{i-1}, P_i)$ and $R(P_i, P_{i+1})$. Note that $P_{k+1} := P_1$ and $\mathcal{F}_{k+1} := \mathcal{F}_1$. To check the existence of such a curve, we construct a digraph $D = (\mathcal{F}, A)$, where

$$\begin{aligned}
\mathcal{F} &:= \bigcup_i \mathcal{F}_i \\
A &:= \{(F_i, F_{i+1}) \mid i \in \{1, \ldots, k\},\ F_i \in \mathcal{F}_i,\ F_{i+1} \in \mathcal{F}_{i+1}, \\
&\qquad \exists H \in \mathcal{H}(r_b, r_p) \text{ inersecting } F_i,\ P_i, \text{ and } F_{i+1}\}.
\end{aligned}$$

Then, finding a hole cut of size k is equivalent to finding a dicycle of length k in D.

With this observation, we can compute MIN-CUT by Algorithm 2, and it is obvious that it runs in polynomial time. □

5 Experimental Results

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

Algorithm 2 Find-MIN-CUT

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

Output: a hole cut of minimum size

```
construct the digraph D = (F, A) defined as above
for all F ∈ F_1 do
    by a breadth-first-search from F, try to find a dicycle of length k in D containing F
    if such a dicycle exists then
        return a hole cut of size k corresponding to the dicycle
    end if
end for
return a hole cut of size k + 1
```

As we have seen before, our algorithm for computing MAX-FLOW consists of the induction step (Algorithm 1) and the base cases (Claim 2). Practically, since most short s-t paths satisfy the second condition of Claim 2, we do not need an implementation of the algorithm in Claim 2. Therefore, we can compute MAX-FLOW by just applying Algorithm 1, repeatedly. We generated input plane graphs with 1000 nodes randomly in a 300×400 rectangular and applied our algortihm to them. Then we can solve the GMFCD in a few seconds. As an example, a computational result with 1000 nodes and 7 paths is shown in Fig. 9, where we set $r_b = 10$ and $r_p = 30$. Note that we regarded the nodes on the boundary of the protective disks as the terminals to make the figure easier to see.

We also implemented Algorithm 2, and applied it to randomly generated graphs. For graphs with 1000 nodes, Algorithm 2 computes MIN-CUT in a few seconds. Fig. 10 is a computational result with 1000 nodes, where we set $r_b = 7$ and $r_p = 30$. In this case, we can see MIN-CUT = MAX-FLOW = 8.

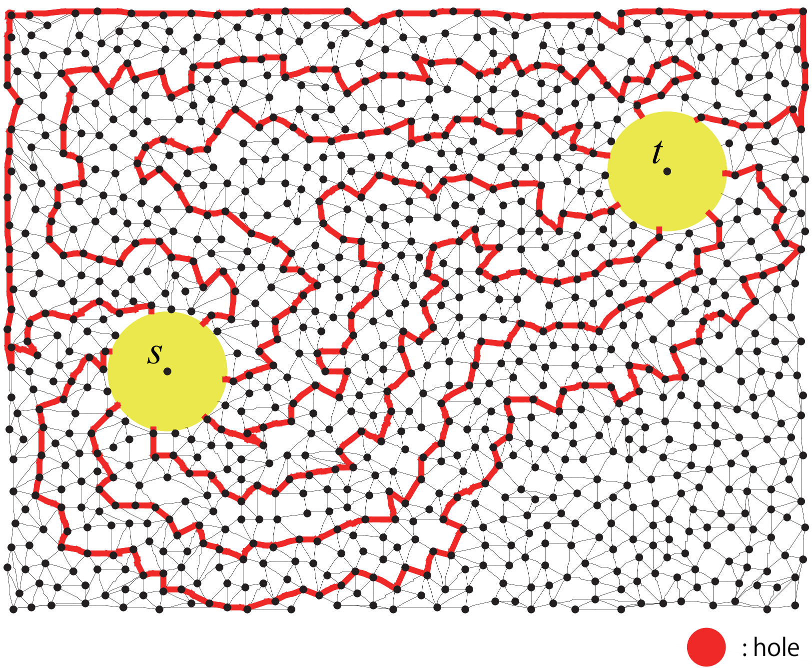

Figure 9: Experimental result (MAX-FLOW)

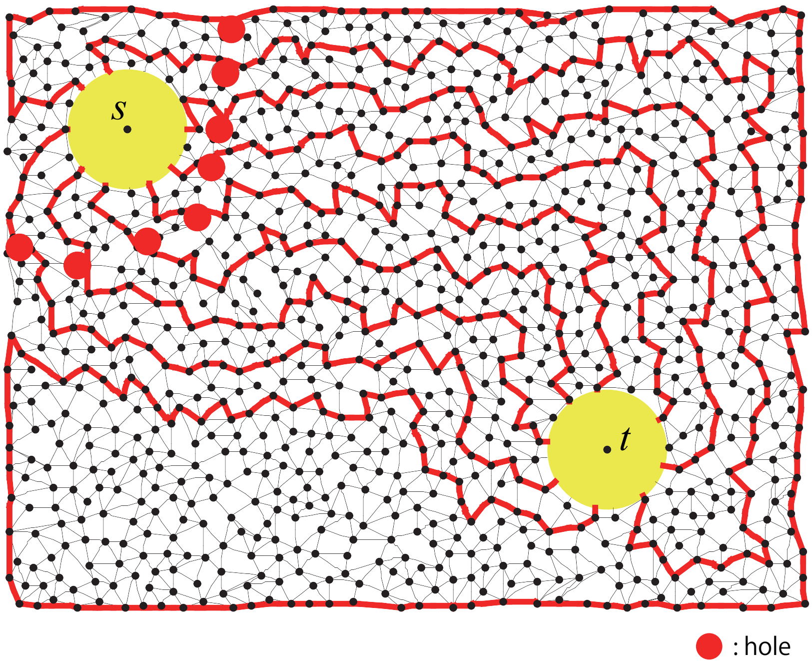

Figure 10: Experimental result (MIN-CUT)

6 Concluding Remarks

In this paper, we discussed the geographical min-cut and the geographical max-flow in the model, in which every hole is a disk of the same radius r_b. We proved a min-max theorem and gave polynomial-time algorithms for the GMFCD and the GMCCD that can be applied to large graphs.

Our results can be extended to the case with holes of different shapes. Suppose that $\mathcal{H}$ is a set of convex shapes (holes) satisfying the following property.

> **Property:** Suppose that two line segments L_1 in $H_1 \in \mathcal{H}$ and L_2 in $H_2 \in \mathcal{H}$ are corssing. Then, H_1 also conatins an endpoint of L_2 or H_2 also conatins an endpoint of L_1. (See the case analysis of the proof of Theorem 3.)

In Theorems 1 and 3, we can replace $\mathcal{H}(r_b, r_p)$ with any set $\mathcal{H}$ satisfying the above property. For example, $\mathcal{H}$ can be a set of disks of different sizes or a set of axis parallel squares. In particular, by setting $\mathcal{H}$ as the set of all edges (not incident to s and t), we obtain a min-max theorem for the maximum induced disjoint s-t paths [6] as a special case of Theorem 1.

Note that we cannot extend our proofs to the case when $\mathcal{H}$ is a set of general connected shapes. Actually, computing MAX-FLOW becomes NP-hard for the case with general $\mathcal{H}$ [2].

References

[1] C. Bettstetter and C. Hartmann: Connectivity of wireless multihop networks in a shadow fading environment, *Wireless Networks*, 11 (2005), 571–579.

[2] D. Bienstock: Some generalized max-flow min-cut problems in the plane, *Mathematics of Operations Research*, 16 (1991), 310–333.

[3] Y.C. Cheng and T.G. Robertazzi: Critical connectivity phenomena in multihop radio models, *IEEE Transactions on Communications*, 37 (1989), 770–777.

[4] P. Gupta and P. Kumar: Critical power for asymptotic connectivity in wireless networks, *Stochastic Analysis, Control, Optimization and Applications*, (1998), 547–566.

[5] C. McDiarmid, B. Reed, A. Schrijver and B. Shepherd: Non-interfering network flows, in *Proceedings of the Third Scandinavian Workshop on Algorithm Theory (SWAT 1992)*, 1992, 245–257.

[6] C. McDiarmid, B. Reed, A. Schrijver and B. Shepherd: Induced circuits in planar graphs, *Journal of Combinatorial Theory, Series B*, 60 (1994), 169–176.

[7] S. Neumayer, A. Efrat and E. Modiano: Geographic max-flow and min-cut under a circular disk failure model, in *Proceedings of the 31st IEEE International Conference on Computer Communications (IEEE INFOCOM 2012)*, 2012, 2736–2740.

[8] S. Neumayer and E. Modiano: Network reliability with geographically correlated failures, in *Proceedings of the 29th IEEE International Conference on Computer Communications (IEEE INFOCOM 2010)*, 2010.

[9] S. Neumayer and E. Modiano: Network reliability under random circular cuts, in *Proceedings of the IEEE Global Telecommunications Conference (IEEE GLOBECOM 2011)*, 2011.

[10] A. Sen, S. Murthy and S. Banerjee: Region-based connectivity - a new paradigm for design of fault-tolerant networks, in *Proceedings of the 15th International Conference on High Performance Switching and Routing (HPSR 2009)*, 2009, 94–100.

[11] A. Sen, B.H. Shen, L. Zhou and B. Hao: Fault-tolerance in sensor networks: A new evaluation metric, in *Proceedings of the 25th IEEE International Conference on Computer Communications (IEEE INFOCOM 2006)*, 2006.