MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Routing Algorithms under Mutual Interference Constraints

Kota ISHIHARA and Yusuke KOBAYASHI

METR 2013–14 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

Routing Algorithms under Mutual Interference Constraints

Kota Ishihara[*†]   Yusuke Kobayashi[*†‡]

August 22, 2013

Abstract

In wireless sensor networks, transportation networks, or VLSI-layout, routing is a fundamental problem and it can be modeled as finding paths with some conditions in a given graph. Among such types of problems, finding disjoint paths connecting given terminal pairs is called the disjoint paths problem, and it is well-studied in the fields of theoretical computer science and graph algorithms. In this paper, we consider a problem of finding paths that are not only disjoint but also "far" from each other, which aims at reducing mutual interference among paths. Our theoretical contribution is to give polynomial-time algorithms for some cases of this problem. We also propose a solution based on the integer programming, which can be applied to many kinds of routing problems.

1 Introduction

The disjoint paths problem is a natural mathematical model of routing problems that appear in wireless sensor networks, transportation networks, VLSI-layout, and so on [3, 15]. In this problem, we are given a graph and its node pairs (s_1, t_1), (s_2, t_2), $\dots$, (s_k, t_k), and the objective is to find disjoint paths $P_1, P_2, \dots, P_k$ such that P_i connects s_i and t_i for $i = 1, 2, \dots, k$. In a wireless sensor network, this problem amounts to sending data from a sensor s_i to another sensor t_i along the path P_i so that each sensor (node) belongs to at most one path. See [9, 18] for related problems motivated by wireless sensor networks. In many practical situations, it is important to consider the case when the given graph is embedded on a two-dimensional plane. For example, the Delaunay triangulation or other planar graphs are often used in routing problems in ad hoc wireless networks [1, 5, 10, 19, 20]. In this paper, we focus on the disjoint paths problem in plane graphs (i.e., graphs embedded on a two-dimensional plane).

The disjoint paths problem is well-studied in the fields of theoretical computer science and graph algorithms, and there are many theoretical results on several

[*]University of Tokyo, Tokyo 113-8656, Japan

[†]JST, ERATO, Kawarabayashi Large Graph Project, Japan

[‡]Supported by Grant-in-Aid for Scientific Research, MEXT, Japan. E-mail: kobayashi@mist.i.u-tokyo.ac.jp

variants of the problem both in general graphs and in plane graphs. When k is a part of the input of the problem, this is one of Karp's NP-complete problems [7], and it remains NP-complete even in plane graphs [11]. Robertson and Seymour [16] gave a polynomial-time algorithm for the disjoint paths problem when the number of terminals, k, is fixed. This algorithm is based on their seminal work on graph minor project, which is spanning 23 papers and giving several deep and profound results and techniques in discrete mathematics. If the input graph is restricted to be planar, the running time is improved to linear time [13, 14].

Although the disjoint paths problem is one of the simplest models of routing problems, we need state-of-the-art techniques in graph algorithms to design a polynomial-time algorithm for the problem. Therefore, it is a challenging task to consider the disjoint paths problem with some additional constraints. In most practical situations of routing problems, it is natural to assume the existence of mutual interference between two paths when they are close to each other (see e.g. [4, 6, 12]). In this paper, we want to find paths that are not only disjoint but also "far" from each other, which aims at reducing mutual interference among paths. More precisely, we consider the following problem

<u>Non-interference Paths Problem</u>

Input: A plane graph $G = (V, E)$ and its node pairs $(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$.

Find: Disjoint paths $P_1, P_2, \dots, P_k$ such that each P_i connects s_i and t_i and $dist(P_i, P_j) > 1$ for distinct $i, j \in \{1, 2, \dots, k\}$ (or conclude that such paths do not exist).

Here, a plane graph G is embedded on a two-dimensional Euclidian space so that each edge is drawn as a line segment, and we are given coordinates of each node. For two points $u, v \in V$, $dist(u, v)$ is defined as a standard Euclidian distance between u and v. In this paper, we adopt the following two definitions of the distance between two paths P_i and P_j.

(1) Regard each path P_i as a subset of the two-dimensional plane and define

$$dist(P_i, P_j) = \min_{u \in P_i, v \in P_j} dist(u, v).$$

(2) Regard each path P_i as a sequence of nodes in V and let $V(P_i) \subseteq V$ be its vertex set. Then, define

$$dist(P_i, P_j) = \min_{u \in V(P_i), v \in V(P_j)} dist(u, v).$$

For example, in Fig. 1, $dist(P_i, P_j) = 0.8$ if we adopt the definition (1) and $dist(P_i, P_j) = 1.2$ if we adopt the definition (2).

Roughly speaking, the first definition models the wired communication in which mutual interference among wires are considered, and the second one deals with the wireless communication in which mutual interference among nodes are taken into consideration. In order to distinguish these two cases, we denote the problem with

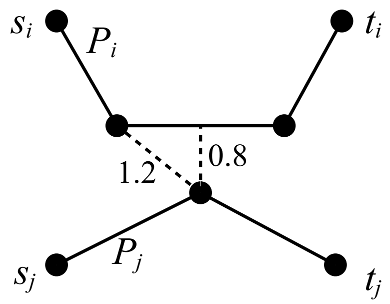

Figure 1: Distance between two paths

the first (resp. second) definition of $dist(P_i, P_j)$ by Non-interference Paths Problem (1) (resp. Non-interference Paths Problem (2)).

When k is a part of the input, both Non-interference Paths Problem (1) and Non-interference Paths Problem (2) are NP-hard, since the disjoint paths problem is NP-hard even in planar graphs [11]. Therefore, in this paper we focus on the case when k is a fixed constant. Our theoretical contributions are as follows.

Theorem 1. *For fixed k, the Non-interference Paths Problem (1) can be solved in polynomial time.*

Theorem 2. *Assume that the plane graph G is the Delaunay triangulation of V. In this case, for fixed k, the Non-interference Paths Problem (2) can be solved in polynomial time.*

Proofs of these theorems are given in Section 2. We also give a solution based on the integer programming, which is discussed in Section 3.

2 Theoretical Results

In this section, we first introduce a problem which generalizes the Non-interference Paths Problem and give a polynomial-time algorithm for it. By using the algorithm, we prove Theorems 1 and 2 in Sections 2.2 and 2.3, respectively.

2.1 Generalized problem

In this subsection, we introduce a variant of the Non-interference Paths Problem, in which we are given a set of node pairs that cannot be contained in different paths. Since the Non-interference Paths Problem will be reduced to this problem in Sections 2.2 and 2.3, we call this problem a *generalized* problem.

Generalized Non-interference Paths Problem (GNPP)

Input: A plane digraph $D = (V, A)$ and its node pairs $(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$. A set of node pairs $N \subseteq V \times V$ with the following property:

(*) for any $(u, v) \in N$, there exists a sequence of nodes $u_0, u_1, \dots, u_{l-1}, u_l$ such that $u_0 = u$, $u_l = v$, $(u_i, u_j) \in N$ for any $0 \leq i < j \leq l$ and either $(u_i, u_{i+1}) \in A$ or $(u_{i+1}, u_i) \in A$ for each $0 \leq i \leq l-1$.

Find: Disjoint directed paths $P_1, P_2, \dots, P_k$ such that each P_i is from s_i to t_i and for any distinct i, j and for any $u \in V(P_i)$ and $v \in V(P_j)$, it holds that $(u, v) \notin N$ (or conclude that such paths do not exist).

The Directed Disjoint Paths Problem in planar digraphs (DDPP) is a special case of this problem, in which $N = \emptyset$ (or $N = \{(v, v) \mid v \in V\}$). In what follows, we give a polynomial-time algorithm for the GNPP based on Schrijver's algorithm for the DDPP [17]. The same approach is also used for the directed *induced* disjoint paths problem in planar digraphs [8].

We now give some preliminaries. A directed edge is called an *arc*, and the nodes $s_1, \dots, s_k, t_1 \dots, t_k$ are called *terminals*. Without loss of generality, we assume that D is weakly connected and each terminal is incident to exactly one arc. Let $\mathcal{F}$ be the set of all faces of D, and $R \in \mathcal{F}$ be the unbounded face of D. For $a \in A$, let left(a) and right(a) be the faces of D at the left-hand side and the right-hand side of a, respectively. The *dual digraph* D^* of D is a digraph $D^* = (\mathcal{F}, A^*)$ whose arc set A^* is defined by $A^* = \{a^* \mid a \in A\}$, where a^* is an arc from left(a) to right(a).

Let $(G_k, \cdot)$ be the free group generated by $g_1, g_2, \dots, g_k$, and let 1 denote its unit element. More precisely, G_k consists of all words $b_1 \cdots b_t$, where $t \geq 0$ and $b_1, \dots, b_t \in \{g_1, g_1^{-1}, \dots, g_k, g_k^{-1}\}$ such that $b_i b_{i+1} \neq g_j g_j^{-1}$ and $b_i b_{i+1} \neq g_j^{-1} g_j$ for $i = 1, \dots, t-1$ and $j = 1, \dots, k$. The product $x \cdot y$ of two words is obtained from the concatenation xy by deleting iteratively all $g_j g_j^{-1}$ and $g_j^{-1} g_j$. A word y is called a *segment* of a word w if $w = xyz$ for certain words x, z. A subset $\Gamma \subseteq G_k$ is called *hereditary* if for each word $y \in \Gamma$ each segment of y belongs to Γ.

We say that a function $\phi : A \to G_k$ is a *flow* if the following three conditions hold.

- For $i = 1, \dots, k$, the arc a leaving s_i satisfies that $\phi(a) = g_i$.
- For $i = 1, \dots, k$, the arc a entering t_i satisfies that $\phi(a) = g_i$.
- For each node $v \in V \setminus \{s_1, \dots, s_k, t_1, \dots, t_k\}$,

  $$\phi(a_1)^{\epsilon_1} \cdot \phi(a_2)^{\epsilon_2} \cdot \dots \cdot \phi(a_l)^{\epsilon_l} = 1,$$

  where $a_1, \dots, a_l$ are the arcs incident with v, in the clockwise order, and $\epsilon_i = +1$ if a_i leaves v and $\epsilon_i = -1$ if a_i enters v.

Note that these conditions correspond to the flow conservation in a standard (single commodity) network flow. We say that two functions $\phi, \psi : A \to G_k$ are *R-homologous* if there exists a function $f : \mathcal{F} \to G_k$ such that

- $f(R) = 1$,
- $f(\text{left}(a))^{-1} \cdot \phi(a) \cdot f(\text{right}(a)) = \psi(a)$ for each arc $a \in A$.

It can be easily seen that if ϕ is a flow and ψ is R-homologous to ϕ, then ψ is also a flow. A flow ψ_Π corresponding to a solution $\Pi = (P_1, \dots, P_k)$ of the GNPP (or the DDPP) is defined by

$$\psi_\Pi(a) = \begin{cases} g_i & \text{if } a \text{ is an arc on } P_i, \\ 1 & \text{otherwise.} \end{cases}$$

Schrijver's algorithm for the directed disjoint paths problem is obtained from Propositions 3 and 4 below.

Proposition 3 (Schrijver [17]). *For each fixed k, we can find in polynomial time a collection of flows $\phi_1, \dots, \phi_N$ with the property that for each solution Π of the DDPP, ψ_Π is R-homologous to at least one of $\phi_1, \dots, \phi_N$.*

Proposition 4 (Schrijver [17]). *There exists a polynomial-time algorithm that, for any flow ϕ, either finds a solution Π of the DDPP such that ψ_Π is R-homologous to ϕ or concludes that such a solution does not exist.*

Proposition 3 implies the following as a corollary, because a solution of the GNPP is also a solution of the DDPP.

Proposition 5. *For each fixed k, we can find in polynomial time a collection of flows $\phi_1, \dots, \phi_N$ with the property that for each solution Π of the GNPP, ψ_Π is R-homologous to at least one of $\phi_1, \dots, \phi_N$.*

In order to design an algorithm for the GNPP, we need the following proposition, which corresponds to Proposition 4. A proof is given in the appendix.

Proposition 6. *There exists a polynomial-time algorithm that, for any flow ϕ, either finds a solution Π of the GNPP such that ψ_Π is R-homologous to ϕ or concludes that such a solution does not exist.*

Now we are ready to give an algorithm for the GNPP.

Theorem 7. *For fixed k, the Generalized Non-interference Paths Problem (GNPP) can be solved in polynomial time.*

Proof. By Proposition 5, we can find a collection of flows $\phi_1, \dots, \phi_N$ such that for each solution Π of the GNPP, ψ_Π is R-homologous to at least one of $\phi_1, \dots, \phi_N$. By Proposition 6, we can either find a solution Π of the GNPP such that ψ_Π is R-homologous to ϕ_i or conclude that such a solution does not exist, for each $i = 1, \dots, N$. Thus we can solve the GNPP in polynomial time when k is a fixed constant. □

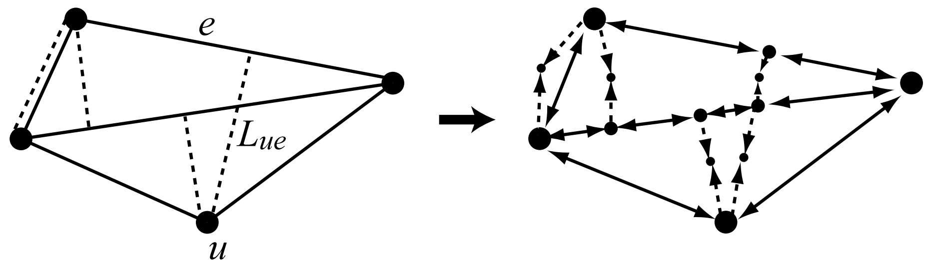

Figure 2: Reduction to the GNPP. (Dotted lines are in E'.)

2.2 Non-interference Paths Problem (1)

In this subsection, by using Theorem 7, we give a proof of Theorem 1, which we restate here.

Theorem. *For fixed k, the Non-interference Paths Problem (1) can be solved in polynomial time.*

Proof. Suppose that we are given an instance of the Non-interference Paths Problem (1), in which each edge is a line segment in the embedding of G. Now we construct an instance of the GNPP that is equivalent to the original instance as follows (see Fig. 2).

For nodes $u, v \in V$, let L_{uv} be the line segment connecting u and v. For a node $v \in V$ and an edge $e \in E$, let L_{ue} be the shortest line segment connecting u and a vertex in e. Note that, if $e = v_1v_2$, then L_{ue} is equal to L_{uv_1}, L_{uv_2}, or the perpendicular line to e. Now we define the following set of line segments:

$$\begin{aligned} E' = \{&L_{uv} \mid u, v \in V,\ \text{length of } L_{uv} \text{ is at most } 1\} \\ &\cup \{L_{ue} \mid u \in V,\ e \in E,\ \text{length of } L_{ue} \text{ is at most } 1\}. \end{aligned}$$

Let $\bar{V} \supseteq V$ be the set of all intersection points of two line segments (edges) in $E \cup E'$. By subdividing every edge in $E \cup E'$ at nodes in $\bar{V}$, we obtain a plane graph $\bar{G} = (\bar{V}, \bar{E} \cup \bar{E}')$, where $\bar{E}$ and $\bar{E}'$ are obtained from E and E', respectively.

Since paths can go through $uv \in \bar{E}$ in either direction, we replace each edge $uv \in \bar{E}$ with two directed arcs (u, v) and (v, u). Similarly, since paths cannot go through $uv \in \bar{E}'$, we replace each edge $uv \in \bar{E}'$ with one new node w and two directed arcs (u, w) and (v, w). Let $D = (V', A)$ be the obtained plane digraph.

Define a set of node pairs $N \subseteq V' \times V'$ by

$$N = \{(u, v) \mid u, v \in V' \text{ on a common line segment in } E'\}.$$

Then, N satisfies the property (*). Furthermore, we can easily see that the obtained instance of the GNPP in D is equivalent to the original instance of the Non-interference Paths Problem (1).

Note that the number of edges in E' is at most $O(|V|^3)$, and so we have $|V'| = O(|V|^6)$, which is a polynomial size of the original instance. Therefore, by Theorem 7, we have a polynomial-time algorithm for the Non-interference Paths Problem (1), which completes the proof of Theorem 1. □

2.3 Non-interference Paths Problem (2)

The Delaunay triangulation of V is the dual of the Voronoi diagram for V. Formally, it is defined as a triangulation of the two dimensional space such that no point in V is inside the circumscribed circle of any triangle in the triangulation. The objective of this subsection is to show Theorem 2, which we restate here.

Theorem. *Assume that the plane graph G is the Delaunay triangulation of V. In this case, for fixed k, the Non-interference Paths Problem (2) can be solved in polynomial time.*

Proof. We construct an instance of the GNPP that is equivalent to the original instance as follows. Replace each edge $uv \in E$ with two arcs (u, v) and (v, u), and define

$$N = \{(u, v) \mid u, v \in V,\ dist(u, v) \leq 1\}.$$

It is easy to see that the obtained instance is equivalent to the original one. Thus, the remaining task is to show that N satisfies the property (*). Although this is one of the basic properties of Delaunay triangulations[1], we give a proof for completeness.

For each node $v \in V$, let R_v be the Voronoi region corresponding to v. Let L be the line segment connecting u and v, where $(u, v) \in N$. Suppose that L traverses Voronoi regions $R_{u_0}, R_{u_1}, \dots, R_{u_l}$ in this order when we walk from u to v (see Fig. 3). Since the Delaunay triangulation of V is the dual of the Voronoi diagram for V, we can see that $u_0 = u$, $u_l = v$, and $u_i u_{i+1} \in E$ for each $0 \leq i \leq l-1$. Furthermore, for each $i = 0, 1, \dots, l$, there exists a point p_i on L such that $dist(p_i, u_i) \leq \min\{dist(p_i, u), dist(p_i, v)\}$, which implies that u_i is inside the circle C whose diameter is L (see Fig. 3). Therefore, $dist(u_i, u_j) \leq dist(u, v) \leq 1$ for any $0 \leq i < j \leq l$, and hence $(u_i, u_j) \in N$ by the definition of N. By the above arguments, N satisfies the property (*).

Therefore, by Theorem 7, we have a polynomial-time algorithm for the Non-interference Paths Problem (2). □

3 Solution via Integer Programming

In the previous section, we discussed theoretical results on the GNPP and the Non-interference Paths Problem. Although the proposed algorithms run in polynomial time, they are too complicated to implement and unlikely to be fast in practice. In this section, we propose two Integer Programming (IP) formulations of the Non-interference Paths Problem to solve the problem in practical time. Since our IP formulations can be applied to both Non-interference Paths Problem (1) and Non-interference Paths Problem (2), we do not distinguish them in most part of this section.

[1] For example, a similar property is used to show that the Delaunay triangulation is a geometric spanner (see [2]).

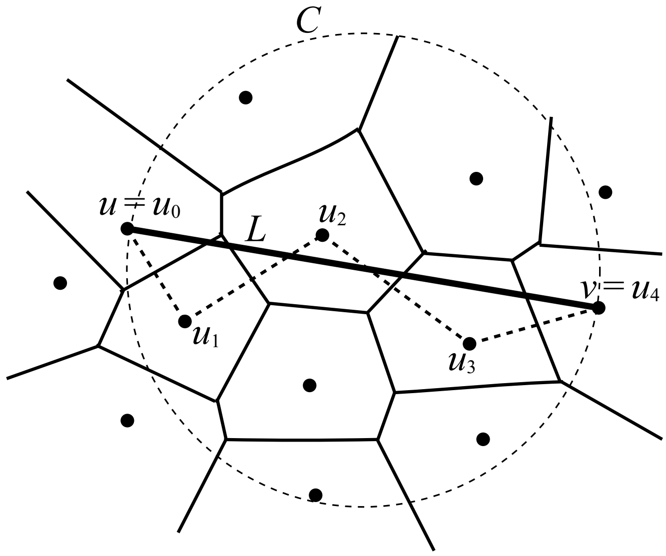

Figure 3: Definition of $u_0, u_1, \ldots, u_l$ and C

3.1 Formulation with Integer Programming

We gave a polynomial-time algorithm for the Non-interference Paths Problem in the previous section, but there are some issues remained:

- Our algorithm is too complicated to implement and time-consuming.
- We want "better" solutions in some sense if there are more than one feasible solutions.
- Given an infeasible instance of the Non-interference Paths Problem, we want to find paths that connect as many (s_i, t_i)-pairs as possible.
- We want a unified approach dealing with variants of the Non-interference Paths Problem, which might have different objective functions, additional constraints, and different definitions of "interference".

We now propose two IP formulations to clear them up. Suppose we are given a plane graph $G = (V, E)$ and its node pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$ as an input of the Non-interference Paths Problem. Although G is undirected, to formulate the problem, we fix a direction of each edge arbitrarily. Then, we can define the head, the tail, the forward direction, and the backward direction of each edge $e \in E$. First, we introduce new parameters that are easily computed from the input:

- $H_{e,v} \in \{-1, 0, 1\}$ $(e \in E, v \in V)$: $+1/-1$ if the head/tail of edge e is v and 0 otherwise.
- $I_{e,e'} \in \{0, 1\}$ $(e, e' \in E)$: 1 if edges e and e' interfere and 0 otherwise.

Here, we say that two edges e and e' *interfere* if $dist(e, e') \leq 1$, i.e. we cannot select two paths P and P' such that P and P' contain e and e', respectively.

Let $[k] = \{1, 2, \ldots, k\}$. In our IP formulations, an s_i-t_i path is regarded as a *flow* from s_i to t_i, and we use the following variables:

(A1) $F_{i,e} \in \{-1, 0, 1\}$ $(i \in [k], e \in E)$: $+1/-1$ if the flow indexed by i goes through edge e in the forward/backward direction and 0 otherwise.

(A2) $\bar{F}_{i,e} \in \{0, 1\}$ $(i \in [k], e \in E)$: absolute value of $F_{i,e}$.

Furthermore, we introduce the following variable for ease of reading:

(A3) $B_{i,v} := \sum_{e \in E} F_{i,e} H_{e,v}$ $(i \in [k], v \in V)$.

In the study of network flows, this value is called a *boundary* at v of the flow indexed by i.

Our first formulation aims at finding a solution with the shortest total length, which is described as follows.

IP formulation (I) of Non-interference Paths Problem

Input: a plane graph $G = (V, E)$ with terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$, $H_{e,v}$ $(e \in E, v \in V)$, and $I_{e,e'}$ $(e, e' \in E)$.

Minimize: $\sum_{i \in [k],\, e \in E} \bar{F}_{i,e}$

Subject to:

(C1) $F_{i,e} \leq \bar{F}_{i,e}$ and $-F_{i,e} \leq \bar{F}_{i,e}$ $\quad(\forall i \in [k],\ \forall e \in E)$

(C2) $B_{i,v} = 0$ $\quad(\forall i \in [k],\ \forall v \in V \setminus \{s_i,\ t_i\})$

(C3) $B_{i,s_i} = -1$ and $B_{i,t_i} = 1$ $\quad(\forall i \in [k])$

(C4) $-1 \leq F_{i,e} + F_{i',e'} \leq 1$ and $-1 \leq F_{i,e} - F_{i',e'} \leq 1$
$\quad(\forall i, i' \in [k]$ with $i \neq i'$, $\forall e, e' \in E$ with $I_{e,e'} = 1)$

and (A1)–(A3).

Since we minimize $\sum_{i,e} \bar{F}_{i,e}$, Constraint (C1) guarantees that $\bar{F}_{i,e}$ coincides with the absolute value of $F_{i,e}$. Constraints (C2) and (C3) mean that $F_{i,e}$ $(e \in E)$ represents a flow from s_i to t_i. Constraint (C4) guarantees that there is no interference among flows i.e. paths connecting terminal pairs. This formulation gives us the shortest feasible solution, which seems to be reasonable in practical applications.

In our second IP formulation, we want to find a maximum number of paths that connect given terminal pairs. Note that this formulation can also be applied to the case where all terminal pairs cannot be connected by non-interference paths.

IP formulation (II) of Non-interference Paths Problem

Input: a plane graph $G = (V, E)$ with terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$, $H_{e,v}$ $(e \in E, v \in V)$, and $I_{e,e'}$ $(e, e' \in E)$.

Maximize: $\sum_{i \in [k]} B_{i,t_i}$

Subject to: (A1), (A3), (C2), (C4), and

$$B_{i,t_i} \in \{0, 1\} \quad (\forall i \in [k]).$$

Note that in this formulation, even optimal solutions might contain unnecessary cycles. In such a case, we can easily derive a solution of the original problem by the breadth-first-search.

The proposed IP formulations of the Non-interference Paths Problem have similar constraints but have distinct objective functions. In practical situations, we can easily modify our formulations to represent the actual objective. We also emphasize here that our IP formulations can represent any interference among edges, which will be useful to deal with practical problems.

3.2 Simulations

We evaluate the performance of our IP formulations (I) and (II) of the Non-interference Paths Problem by computational experiments. For experiments, we randomly generated a plane graph with 270 nodes in a 20×20 square area, and chose k $(2 \leq k \leq 8)$ terminal pairs randomly. Two edges e and e' interfere if $dist(e, e') \leq 1$ in the sense of the definition (1), that is, we consider the Non-interference Paths Problem (1). We solve IP instances with mathematical programming solvers IBM ILOG CPLEX 12.5 and NUOPT 15.1.0[2]. Our experiments were conducted on the computer with Intel Xeon 3.20 GHz (4 cores) and 16GB of memory.

For both IP formulations, the running time is heavily depending on the arrangement of the terminals. Roughly, we can solve instances with four or less terminal pairs quickly (less than ten minutes), and we require a few hours to solve instances of six terminals. We remark here that in most cases we can find a good feasible solution quickly, and it takes a long time to show the optimality of the solution. In Fig. 4, we show a solution of a case of $k = 7$ obtained by using the IP formulation (I). An instance shown in Fig. 5 has eight terminal pairs, but we cannot connect all the terminal pairs by non-interference paths. By using a modification of the IP formulation (II), we found six non-interference paths connecting terminals, where stars and squares represent terminal pairs that were not connected by non-interference paths.

4 Conclusions

In this paper, we introduced the Non-interference Paths Problem as a natural extension of the disjoint paths problem. We gave polynomial time algorithms for

[2]Product of Mathematical Systems Inc. (http://www.msi.co.jp/english/)

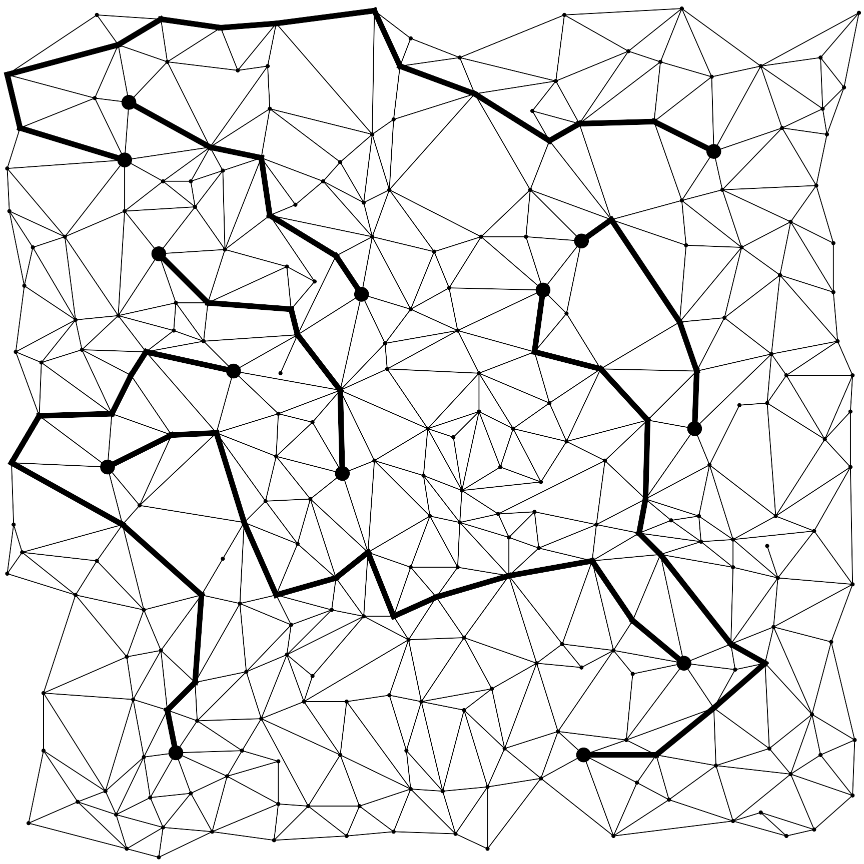

Figure 4: Experimental result by IP formulation (I)

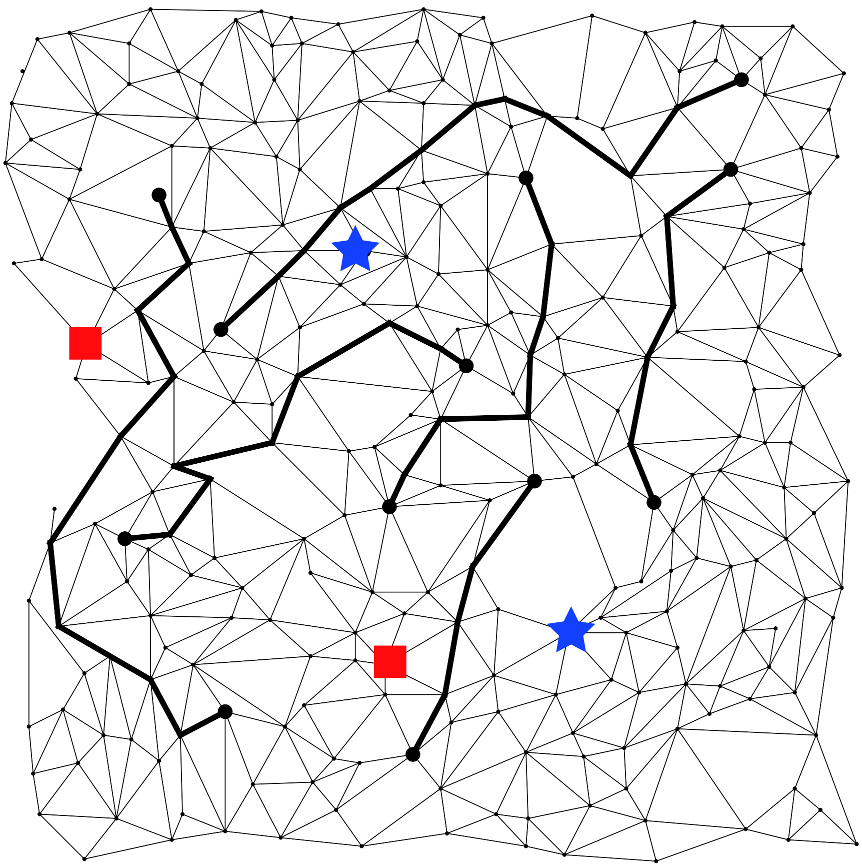

Figure 5: Experimental result by IP formulation (II)

some cases of this problem, which are theoretically interesting but far from practical use. It is open whether Theorem 2 can be extended to the case when G is a general plane graph.

We also solved the Non-interference Paths Problem by using IP formulations, and evaluated the performance of this approach. With this approach, we can deal with many kinds of objective functions and constraints, and we can solve small instances efficiently. If we need to solve larger instances, heuristic methods should be adopted instead of the IP formulations.

Acknowledgement

The authors thank Kei Kimura for helpful comments.

References

[1] P. Bose, P. Morin, I. Stojmenović and J. Urrutia: Routing with guaranteed delivery in ad hoc wireless networks, *Wireless Networks*, 7 (2001), 609–616.

[2] K.J.S. David P. Dobkin, Steven J. Friedman: Delaunay graphs are almost as good as complete graphs, *Discrete and Computational Geometry*, 5 (1990), 399–407.

[3] A. Frank: Packing paths, cuts and circuits — a survey, in B. Korte, L. Lovász, H.J. Prömel and A. Schrijver, eds., *Paths, Flows, and VLSI-Layout*, Springer-Verlag, 1990, 49–100.

[4] P. Gupta and P.R. Kumar: The capacity of wireless networks, *IEEE Transactions on Information Theory*, 46 (2000), 388–404.

[5] M.B. Haider, S. Imahori and K. Sugihara: Success guaranteed routing in almost delaunay planar nets for wireless sensor communication, *International Journal of Sensor Networks*, 9 (2011), 69–75.

[6] M.M. Halldórsson and P. Mitra: Wireless capacity and admission control in cognitive radio, in *Proceedings of the 31st IEEE International Conference on Computer Communications (IEEE INFOCOM 2012)*, 2012, 855–863.

[7] R.M. Karp: On the computational complexity of combinatorial problems, *Networks*, 5 (1975), 45–68.

[8] Y. Kobayashi: Induced disjoint paths problem in a planar digraph, *Discrete Applied Mathematics*, 157 (2009), 3231–3238.

[9] V.S.A. Kumar, M.V. Marathe, S. Parthasarathy and A. Srinivasan: Algorithmic aspects of capacity in wireless networks, in *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 2005, 133–144.

[10] X.Y. Li, G. Calinescu, P.J. Wan and Y. Wang: Localized Delaunay triangulation with application in ad hoc wireless networks, *IEEE Transactions on Parallel and Distributed Systems*, 14 (2003), 1035–1047.

[11] J.F. Lynch: The equivalence of theorem proving and the interconnection problem, *SIGDA Newsletter*, 5 (1975), 31–36.

[12] S. Ramanathan and E.L. Lloyd: Scheduling algorithms for multihop radio networks, *IEEE/ACM Transactions on Networking*, 1 (1993), 166–177.

[13] B. Reed: Rooted routing in the plane, *Discrete Applied Mathematics*, 57 (1995), 213–227.

[14] B. Reed, N. Robertson, A. Schrijver and P.D. Seymour: Finding disjoint trees in planar graphs in linear time, in *Contemporary Mathematics 147*, American Mathematical Society, 1993, 295–301.

[15] N. Robertson and P.D. Seymour: An outline of a disjoint paths algorithm, in B. Korte, L. Lovász, H.J. Prömel and A. Schrijver, eds., *Paths, Flows, and VLSI-Layout*, Springer-Verlag, 1990, 267–292.

[16] N. Robertson and P.D. Seymour: Graph minors. XIII. the disjoint paths problem, *Journal of Combinatorial Theory, Series B*, 63 (1995), 65–110.

[17] A. Schrijver: Finding k disjoint paths in a directed planar graph, *SIAM Journal on Computing*, 23 (1994), 780–788.

[18] A. Srinivas and E. Modiano: Finding minimum energy disjoint paths in wireless ad-hoc networks, *Wireless Networks*, 11 (2005), 401–417.

[19] Y. Wang and X.Y. Li: Efficient Delaunay-based localized routing for wireless sensor networks: Research articles, *Int. J. Commun. Syst.*, 20 (2007), 767–789.

[20] C.H. Wu, K.C. Lee and Y.C. Chung: A Delaunay triangulation based method for wireless sensor network deployment, *Comput. Commun.*, 30 (2007), 2744–2752.

A Proof of Proposition 6

In order to show Proposition 4, Schrijver [17] introduced a new problem called cohomology feasibility problem (CFP), and gave a polynomial-time algorithm for it. He showed that Proposition 4 can be derived from the polynomial-time algorithm for the CFP. In the appendix, we describe the CFP and show that Proposition 6 can also be obtained from the polynomial-time algorithm for the CFP. We note that almost the same argument is used in [8].

Let $D = (V, A)$ be a weakly connected digraph, which may have parallel arcs, and let $r \in V$. Two functions $\phi, \psi : A \to G_k$ are called *r-cohomologous* if there exists a function $f : V \to G_k$ such that

- $f(r) = 1$,
- $\psi(a) = f(u)^{-1} \cdot \phi(a) \cdot f(v)$ for each arc $a = (u, v) \in A$.

Schrijver introduced the following problem called *cohomology feasibility problem (CFP)*, and showed that it can be solved in polynomial time.

Cohomology Feasibility Problem (CFP)

Input: A weakly connected digraph $D = (V, A)$, a node $r \in V$, a function $\phi : A \to G_k$, a hereditary subset $\Gamma(a) \subseteq G_k$ for each arc $a \in A$.

Find: A function $\psi : A \to G_k$ such that ψ is r-cohomologous to ϕ and $\psi(a) \in \Gamma(a)$ for each arc $a \in A$ (or conclude that such a function does not exist).

Theorem 8 (Schrijver [17])**.** *The CFP can be solved in polynomial time of* $|A|$*,* σ*, and* k*, where* $\sigma = \max\{|\Gamma(a)| \mid a \in A\}$.

We now show Proposition 6 by using this theorem.

Let $D^* = (\mathcal{F}, A^*)$ be the dual digraph of D. Let A_1 be the set of all chords in all faces of D^*. More precisely, we consider all nonadjacent node pairs $F, F' \in \mathcal{F}$ which are on the boundary of a face of D^*, and define A_1 as the set of all arcs $a_{F,F'}$ from F to F'.

For each $(u, v) \in N$, we take a sequence $u_0, u_1, \ldots, u_l \in V$ as in the property (*), and we consider the digraph $D^*_{(u,v)} := D^* - \{(u_0, u_1), (u_1, u_2), \ldots, (u_{l-1}, u_l)\}$. Then, $l+1$ faces of D^* each containing u_i make up a new face $w_{(u,v)}$ of $D^*_{(u,v)}$. Let $A_{(u,v)}$ be the set of all chords in $w_{(u,v)}$ which are not in $A^* \cup A_1$, and let $A_2 = \bigcup_{(u,v) \in N} A_{(u,v)}$. We construct a new digraph $D^+ = (\mathcal{F}, A^+)$, where $A^+ = A^* \cup A_1 \cup A_2$.

We define $\phi^+ : A^+ \to G_k$ as follows:

- $\phi^+(a^*) = \phi(a)$ for each arc $a \in A$.
- For each $a_{F,F'} \in A_1 \cup A_2$, let $\pi = ((a_1^*)^{\epsilon_1}, (a_2^*)^{\epsilon_2}, \ldots, (a_l^*)^{\epsilon_l})$ be the dipath traveling clockwise from F to F' on the boundary of the face of D^* or $D^*_{(s,t)}$, where $\epsilon_i \in \{+1, -1\}$. Then $\phi^+(a_{F,F'}) = \phi(a_1)^{\epsilon_1} \cdot \phi(a_2)^{\epsilon_2} \cdot \cdots \cdot \phi(a_l)^{\epsilon_l}$.

We say that ϕ^+ is the *extended function* of ϕ.

For each arc $a' \in A^+$, we define $\Gamma^+(a') \subseteq G_k$ as follows:

- $\Gamma^+(a') = \{1, g_1, \ldots, g_k\}$ for $a' \in A^*$,
- $\Gamma^+(a') = \{1, g_1, g_1^{-1}, \ldots, g_k, g_k^{-1}\}$ for $a' \in A_1$, and
- $\Gamma^+(a') = \{1, g_i^t, g_i^{-t} \mid i = 1, \ldots, k,\ t = 1, \ldots, n\}$ for $a' \in A_2$.

Then finding a solution Π of the GNPP in D such that ψ_Π is R-homologous to ϕ corresponds to solving the CFP in D^+ with respect to ϕ^+ and Γ^+. We now show this fact.

Suppose that $\psi_\Pi : A \to G_k$ corresponds to a solution Π of the GNPP which is R-homologous to ϕ. Then we can see that its extended function $\psi_\Pi^+ : A^+ \to G_k$ is

R-cohomologous to ϕ^+. Since no pair of dipaths in Π have common arcs or common nodes, we have $\psi^+_\Pi(a') \in \Gamma^+(a')$ for any $a' \in A^* \cup A_1$. Furthermore, for a sequence $u_0, u_1, \ldots, u_l$ corresponding to a pair $(u, v) \in N$, at most one dipath in Π can go through nodes in $\{u_0, u_1, \ldots, u_l\}$, which shows that $\psi^+_\Pi(a') \in \Gamma^+(a')$ for any $a' \in A_2$. Hence, ψ^+_Π is a solution of the CFP.

Conversely, suppose that $\psi^+ : A^+ \to G_k$ is a solution of the CFP. Define $\psi : A \to G_k$ by $\psi(a) = \psi^+(a^*)$ for each $a \in A$. Then we can see that ψ is R-homologous to ϕ and ψ^+ is the extended function of ψ. For each $i = 1, \ldots, k$, define $P_i = \{a \in A \mid \psi(a) = g_i\}$. Since ψ is a flow and $\psi^+(a') \in \Gamma^+(a')$ for any $a' \in A^*$, P_i consists of a dipath from s_i to t_i and some dicycles. Hence, we may assume that P_i is a dipath from s_i to t_i, and $P_1, \ldots, P_k$ are arc-disjoint by the definition of P_i. We now show that $\Pi = (P_1, \ldots, P_k)$ is node-disjoint and $(u, v) \notin N$ if u and v are contained in different dipaths.

Assume that two dipaths P_i and P_j have a common node v for some distinct i, j. Then there exist arcs a_1 and a_2 of D such that both a_1 and a_2 are incident to v, $\psi(a_1) \in \{g_i, g_i^{-1}\}$, and $\psi(a_2) \in \{g_j, g_j^{-1}\}$. Let π be the dipath in D^* whose first and last arcs are a_1^* and a_2^*, respectively, along the boundary of the face of D^* corresponding to v. We may assume that we have chosen a_1 and a_2 such that π is as short as possible. Then $g_i^{\pm 1}$ and $g_j^{\pm 1}$ are segments of $\psi^+(a_{F,F'})$ for an arc $a_{F,F'} \in A_1$, where π is the dipath from F to F', which contradicts the assumption that ψ^+ is a solution of the CFP. Hence, no pair of Π have common nodes.

Assume that P_i has a node u, P_j has a node v, and $(u, v) \in N$ for some distinct i, j. Let $u_0, u_1, \ldots, u_l$ be the sequence corresponding to (u, v). By choosing (u, v) so that the length of this sequence is minimum, we may assume that $u_1, \ldots, u_{l-1}$ are not contained in $P_1, \ldots, P_k$.

We now take two arcs a_1 and a_2 of D such that a_1 is incident to u, a_2 is incident to v, $\psi(a_1) \in \{g_i, g_i^{-1}\}$, and $\psi(a_2) \in \{g_j, g_j^{-1}\}$. Let π be the dipath in D^* whose first and last arcs are a_1^* and a_2^*, respectively, along the boundary of the face of $D^*_{(s,t)}$. We may assume that we have chosen a_1 and a_2 such that π is as short as possible. Then $g_i^{\pm 1}$ and $g_j^{\pm 1}$ are segments of $\psi^+(a_{F,F'})$ for an arc $a_{F,F'} \in A_2$, where π is the dipath from F to F', which contradicts the assumption that ψ^+ is a solution of the CFP.

By the above arguments and Theorem 8, we can find a solution Π of the GNPP such that ψ_Π is R-homologous to ϕ in polynomial time by solving the CFP.