AN ABSTRACT OF THE THESIS OF

Farzad Zafarani for the degree of Master of Science in Computer Science presented on May 20, 2016.

Title: A Structural Theorem For Shortest Vertex-Disjoint Paths Computation in Planar Graphs

Abstract approved: ______________________________________

Amir Nayyeri

Given k terminal pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$ in an edge-weighted graph G, the k Shortest Vertex-Disjoint Paths problem is to find a collection $P_1, P_2, \ldots, P_k$ of vertex-disjoint paths with minimum total length, where P_i is an s_i-to-t_i path. As a special case of the multi-commodity flow problem, computing vertex disjoint paths has found several applications, for example in VLSI design, or network routing.

In this thesis we describe a Structural Theorem for a special case of the Shortest Vertex-Disjoint Paths problem in undirected planar graphs where the terminal vertices are on the boundary of the outer face. At a high level, our Structural Theorem guarantees that the i^{th} path of the k Shortest Vertex-Disjoint paths does not cross j^{th} $(j \neq i)$ path of the $k-1$ Vertex-Disjoint Paths problem.

A Structural Theorem For Shortest Vertex-Disjoint Paths Computation in Planar Graphs

by

Farzad Zafarani

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented May 20, 2016
Commencement June 2017

Master of Science thesis of Farzad Zafarani presented on May 20, 2016.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Farzad Zafarani, Author

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere appreciation and gratitude to my advisor and co-advisor, Professor Amir Nayyeri and Professor Glencora Borradaile for all their helps and guidance that have given me over the past two years. I have learned a lot from both of them during the research meetings and courses that took with them. Also, I would like to thank Professor Eugene Zhang for accepting to be on my committee.

Friendship is the most precious thing we have in our lives and my friends play an important role in my life. I am very thankful to each and every one of them, particularly, my roommate, Reza Ghaeini.

Finally, I would like to thank my lovely parents for their supports.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

Modeling problems using graphs originated in 1735, when Leonhard Euler solved the famous Konigsberg bridge problem. After then, researcher tried to model many real-world problem using graphs. As an example we could mention finding shortest path between two places and Hamiltonian cycle. Till now, it is not known nor believed that all real world problems modeled on graph can be solved efficiently. In other words, some problems are NP-Hard on general graphs. As an example, we present the Minimum Cost Multi-commodity Flow problem as our motivation for this thesis. In this problem you are given a flow graph and k terminals, where you should send different commodities between each pair of terminals and you should minimize the cost of sending flow. The Multi-commodity Flow problem has an application in Communication Networks, Computer Networks, Railway Networks, and Distribution Networks.

In some cases we restrict ourselves to a class of graphs that we can find the optimal solution more efficiently. As an example could be *trees, planar graphs*, and *bounded tree-width* graphs. In this thesis, we will restrict ourselves to the class of planar graphs. As an example of some problems that we can solve much faster in Planar Graphs compared to general graphs we can mention ([5], [14] and [4]).Planar graphs originated with the studies of polytopes and maps and since it is very similar to the real world maps, it has a lot of interesting applications.

By restricting ourselves to the class of planar graphs, we will define the Shortest Vertex-Disjoint Paths problem which is a special case of Multi-Commodity Flow problem and will propose a Structural Theorem for solving the problem when the terminals are on the boundary of the outer face of a planar graph.

This thesis is organized as follows: First, we start with a relevant background in Chapter 2 where we formally define the terminology of graphs, planar graphs, bounded tree-width graphs. We will define the Multi-Commodity flow problem and Steiner Trees in planar graphs. We will briefly go through three different types of path routing problems: Vertex-Disjoint Paths, Edge-Disjoint Paths, and Non-Crossing Paths and will give the known results on each of these problems. In Chapter 3, we will define the Structural Theorem and will use it to solve the Shortest Vertex-Disjoint Paths problem in planar graphs. Finally, in Chapter 4, we conclude with a list of open problems and potential extension for future work.

Chapter 2: Background

A graph G is defined as a pair (V, E) where V is the set of vertices and E is the set of edges. An undirected edge e between vertex u and vertex v is denoted by $\{u, v\}$. For the directed case we denote an edge between vertex u and v by (u, v). Here, u and v are called endpoints of the edge $e = (u, v)$. An embedding of a graph $G = (V, E)$ to $\mathbb{R}^2$ is a mapping from V to distinct points of $\mathbb{R}^2$ and collection of mappings from E to paths of $\mathbb{R}^2$ such that they are disjoint except at their endpoints[22]. A *face* of a graph is a connected component of $\mathbb{R}^2 \backslash E$. A graph $G = (V, E)$ is called *planar* if it can be embedded on the plane. In other words, it can be drawn on the plane such that its edges intersect only at their endpoints.

2.0.1 Graph Minors

A undirected graph H is called a *minor* of the graph G, if H can be obtained from G using a sequence of two operations:

- Deleting edges or vertices
- Contracting Edge: Contracting an edge $e = (u, v)$ removes the edge e while simultaneously merge u and v (See Figure 2.0.1).

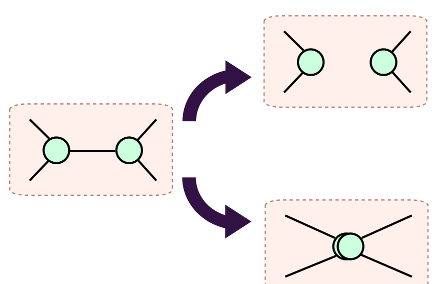

Figure 2.1: Two operations of the graph minors, the top operation is deleting an edge while the bottom operation contracts an edge

Wagner's theorem states that a graph is planar, if an only if it does not include $K_{3,3}$ or K_5 as minor.

Euler proposed a formula on planar graphs that relates the number of vertices and edges to the face.

Theorem 2.0.1. *Let $G = (V, E)$ be a connected planar graph, and let F be the set of faces of G in some planar embedding. Then,* $|V| - |E| + |F| = 2$

Many interesting discrete optimization problems are NP-hard. Thus, unless P = NP, there are no efficient algorithms for solving them. However, by using *Approximation algorithms* we try to find a solution that closely approximates the optimal solution in terms of its value.

Definition 2.0.2. *An α-approximation algorithm for an optimization problem is a polynomial-time algorithm that for all instances of the problem produces a solution whose value is within a factor of α of the value of an optimal solution.*

2.1 Multi-commodity Flow Problem

Give a flow network $G = (V, E)$, where each edge $e \in E$ has a capacity $c(u, v)$. There are k commodities $K_1, K_2, \dots, K_k$ where $K_i = (s_i, t_i, d_i)$ is a tuple where we want to send a demand of d_i from vertex s_i to t_i. The flow of commodity i along an edge (u, v) is $f_i(u, v)$. We should find an assignment such that it satisfies the following three condition:

- Capacity constraints: $\sum_{i=1}^{k} f_i(u, v) \leq c(u, v)$
- Flow conservation: $\sum_{v \in V} f_i(u, v) = 0$ when $u \neq s_i, t_i$
- Demand satisfaction $\sum_{w \in V} f_i(s_i, w) = \sum f_i(w, t_i) = d_i$

In the optimization version of the problem (minimum cost multi-commodity flow problem), there is a cost $a(u, v).f(u, v)$ for sending the flow along the edge (u, v). The goal is to minimize the cost function:

$$\sum_{(u,v) \in E} (a(u, v) \sum_{i=1}^{k} f_i(u, v)) \tag{2.1}$$

Steiner Tree: Given a graph G and a set of vertices $H \subseteq V$, the Steiner Tree problem is to connect vertices in H with minimum cost tree. The steiner tree is very similar to spanning tree problem but it is different in the case that not all of the vertices of the graph should be connected which makes the problem to become NP-hard.
Steiner Forest: Given a graph G and a set of vertices $H_1, H_2, \ldots, H_m$, connect each set with minimum steiner tree.

Tree-width: The treewidth of an undirected graph G is a number that, intuitively, measures the similarity of G to a tree. A width of a tree decomposition is the defined as the size of the largest bag - 1. Treewidth is defined as the smallest width of any tree decomposition of G. A tree decomposition of a graph $G = (V, E)$ is a tree, T, with nodes $X_1, \ldots, X_n$, where each X_i is a subset of V, satisfying the following properties

- The union of all sets X_i equals V. That is, each graph vertex is contained in at least one tree node.
- If X_i and X_j both contain a vertex v, then all nodes X_k of the tree in the (unique) path between X_i and X_j contain v as well. Equivalently, the tree nodes containing vertex v form a connected subtree of T.
- For every edge (v, w) in the graph, there is a subset X_i that contains both v and w. That is, vertices are adjacent in the graph only when the corresponding subtrees have a node in common.

2.2 Path Routing Problems

Here, we define three path routing problems: Vertex-Disjoint Paths, Edge-Disjoint Paths, and Non-Crossing Paths.

2.2.1 Vertex-Disjoint Paths

Two paths P_1, P_2 are vertex-disjoint if they do not share a vertex edge. (See Figure 2.2).

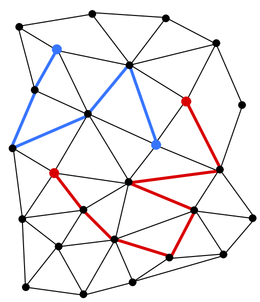

Figure 2.2: Two paths of color blue and red are shown where they are vertex-disjoint
.

2.2.1.1 k Vertex-Disjoint Paths problem

Given graph G and k pairs of terminals $\{(s_1, t_1), \ldots, (s_{,}, t_{,})\}$ the Vertex-Disjoint Paths problem ask for finding k disjoint paths connecting s_i to t_i (For $1 \leq i \leq k$).
The vertex disjoint path is a special case of the multi-commodity flow problem and it has several applications including VLSI design[20], resource allocation, and network routing[23][26].
If k is part of the input, It is one of Karp's NP-hard problems even for undirected planar graphs[16].
However, there are polynomial time algorithms if k is a constant for general undirected graph [25]. In general directed graphs, the k-vertex-disjoint paths problem is NP-hard even for $k = 2$ [11] but is fixed parameter tractable with respect to parameter k in directed planar graphs[9].

2.2.1.2 Shortest Vertex-Disjoint Paths Problem

Given graph G and k pairs of terminals $\{(s_1, t_1), \ldots, (s_k, t_k)\}$, the Shortest Vertex-Disjoint Paths problem (a.k.a min-sum) ask for finding k disjoint paths connecting s_i to t_i (For $1 \leq i \leq k$) with minimum total length

As an example, the 2-min-sum problem and the 4-min-sum problem are open in directed and undirected planar graphs, respectively, even when the terminals are on a common face; neither polynomial-time algorithms nor hardness results are known for these problem. Bjorklund and Husfeldt gave a randomized polynomial time algorithm for the min-sum two vertex-disjoint paths problem in general undirected graphs[2].

One of a few results in this context is due to Colin de Verdière and Schrijver: a polynomial time algorithm for the k-min-sum problem in a (directed or undirected) planar graph, given all sources are on one face and all sinks are on another face [31]. In the same paper, they ask about the existence of a polynomial time algorithm provided all the terminals (sources and sinks) are on a common face.
If the sources and sinks are ordered so that they are in the order $s_1, s_2, \ldots, s_k, t_k, t_{k-1}, \ldots, t_1$ around the boundary, then the k-min-sum problem can be solved by finding a min-cost flow from $s_1, s_2, \ldots, s_k$ to $t_k, t_{k-1}, \ldots, t_1$. For $k \leq 3$ in undirected planar graphs with the terminals in arbitrary order around the common face, Kobayashi and Sommer give an $O(n^4 \log n)$ algorithm. [18]
Borradaile, Nayyeri, and Zafarani proposed $O(kn^5)$ algorithm for the case where all terminals are on the boundary of the outer face of the planar graph and have a specific order which alternate between source and sink[6].

In Table 2.2.1.2 we have expand the Table 2 proposed by Kobayashi and Sommer [18] for vertex-disjoint paths problem.

2.2.1.3 The Min-Max problem

Another similar optimization problem to Shortest Vertex-Disjoint Paths problem is the Min-Max problem. In this problem, you are given a graph G and k pairs of terminals $\{(s_1, t_1), \ldots, (s_k, t_k)\}$, the Min-Max problem tries to find k disjoint paths connecting s_i to t_i (For $1 \leq i \leq k$) such that the length of the longest path is to be minimized. The Min-Max problem seems to be harder than the Min-Sum problem[18]. There are few known results on the min-max problem. Itai, Perl, and Shiloach proved that it is NP-hard on directed acyclic graphs even for two terminals [15]. Li, McCormick, and Simchi-Levi proposed a 2-approximation algorithm for directed graphs with two pairs of terminals[21].
It has been showed that this problem is strongly NP-hard for the directed graphs even

Table 2.1: Shortest Vertex-Disjoint Paths Summary Results

	Conditions	Complexity
k=2	directed	NP-hard
	directed, planar, one face	OPEN
	undirected	OPEN
	undirected, planar, two faces	P
	undirected	Randomized P
k=3	undirected, planar, one face	P
k: fixed	undirected	OPEN
k: general	undirected	NP-hard
	$s_1 = \ldots = s_k$ and/or $t_1 = \ldots = t_k$	P
	planar, one face, well-ordered	P
	planar, $S \neq T$ faces	P
	planar, one face, $(s_1, t_1), ..., (s_k, t_k)$	P
	bounded tree-width	$O(n)$

for 2 terminals[21]. For the undirected planar graphs with *tree-width*$= 3$, it has been proved that it is NP-hard [32][18]. Kobayashi and Sommer have proved that on undirected graphs with *tree-width*≤ 2, the problem is solvable in polynomial time[18].

2.2.2 Edge Disjoint Paths

Two paths P_1, P_2 are edge disjoint if they do not share a common edge. (See Figure 2.3).

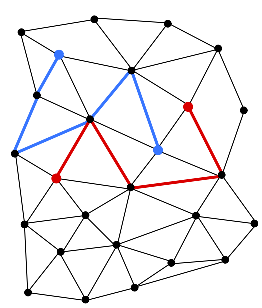

Figure 2.3: Two paths of color blue and red are shown where they are edge-disjoint

.

2.2.2.1 k Edge-Disjoint Paths problem

We are given a graph G and a set of source-sink pairs in G. The goal is connect as many pairs as possible in an edge-disjoint manner. The problem has been proved to be NP-hard. There are lot of progress on this problem in terms of hardness.
The best approximation algorithm so far is $O(min\{n^{2/3},\sqrt{m}\})$ for both directed and undirected graph where n is the number of vertices of the graph and m is the number of edges [7] [17] [19] [27] [30]. In a special case where the graph is acyclic the approximation ratio is improved to $O(\sqrt{n})$ [8].
Similarly, Guruswaami obtained a hardness results of $\Omega(m^{1/2-\epsilon})$ [12]. Andrews and Zhang proved a lower-bound $\Omega(\log^{1/3-\epsilon} n)$ for undirected version [1]. It is NP-hard in directed graphs even for two pair of source and sink. However, in undirected graphs, is solvable in polynomial time for constant number of terminals.
There exist poly-logarithmic algorithm for the special class of graphs including trees, grids, and expanders.

2.2.3 Non-Crossing Paths

Two paths P_1 and P_2 cross if there exist a common path P' in P_1 and P_2 such that if you contract P' to a vertex, the order of edges alternate between the edges of P_1 and P_2 around e in the planar embedding. (For more details see Chapter 1 of [3]).

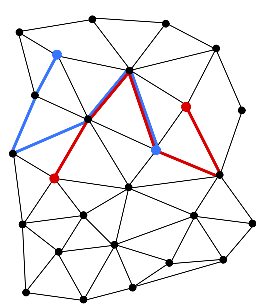

Figure 2.4: Two Non-Crossing paths of color blue and red are shown
.

It has an application in VLSI design [28]. Takahashi et. al. proposed an $O(n \log k)$-

algorithm that finds shortest non-crossing walks in a planar graph, when all terminals lie on at most two obstacle faces. Papadopoulou gave a geometric formulation of the shortest non-crossing walks [24]. Takahashi et. al. proposed an $O(n \log n)$-time algorithm for a rectlinear variant of the geometric problem, where the domain is a rectangle with many rectangular holes and the terminals lie either on the outer boundary or on the boundary of one hole [29]. Erickson and Nayyeri gave a $2^{O(h^2)} n \log k$ fixed parameter tractable algorithm with respect to parameter h(the number of obstacles) for the shortest non-crossing walks in plane graphs [10].

Chapter 3: The Structural Theorem

In this section, we show that two constrained solutions to the disjoint paths problem interact in a limited way under the Uniqueness Assumption. To do so, we compare a solution connecting $k-1$ terminals $\{(s_1,t_1),\ldots,(s_{k-1},t_{k-1})\}$ to a solution connecting k terminals $\{(s_1,t_1),\ldots,(s_k,t_k)\}$.

Shortest disjoint paths problem Let ST $=\{(s_1,t_1),\ldots,(s_k,t_k)\}$ be a set of terminal pairs, let ∂G be the outerface of the planar embedding of the graph G. We wish to find a set of simple paths $\{Q_1,Q_2,\ldots,Q_k\}$ with the following properties:

1. For each $1\le i\le k$, the path Q_i is an (s_i,t_i)-path.

2. For each $1\le i<j\le k$, the path Q_i and Q_j are disjoint.

3. The total length of the paths, $\sum_{1\le i\le k}|Q_i|$ is minimized.

Note that such a set of paths may not exist for a particular order of vertices on the ∂G. We denote this optimization problem by SDPP(ST).

Theorem 3.0.1 (Structure Theorem)**.** *Let* $\mathrm{OPT}_{k-1}=\{Q_1,Q_2,\ldots,Q_{k-1}\}$ *and* $\mathrm{OPT}_k=\{\bar{Q}_1,\bar{Q}_2,\ldots,\bar{Q}_k\}$ *be the optimal solutions to* $\mathrm{SDPP}(\{(s_1,t_1),\ldots,(s_{k-1},t_{k-1})\})$ *and* $\mathrm{SDPP}(\{(s_1,t_1),\ldots,(s_k,t_k)\})$*, respectively. For* $1\le i\le k-1$*,* Q_i *does not cross* $\bar{Q}_j$ *for* $j\ne i$*.*

In the remainder of this section, we prove the Structure Theorem. First we give a bird's-eye view of the proof. We modify OPT_k and OPT_{k-1} to obtain two new sets of disjoint paths: $\mathrm{OPT}'_k=\{\bar{P}_1,\bar{P}_2,\ldots,\bar{P}_k\}$ and $\mathrm{OPT}'_{k-1}=\{P_1,P_2,\ldots,P_{k-1}\}$ with the following properties:

1. $\bar{P}_i$ is an (s_i,t_i)-path for $1\le i\le k$ and P_i is an (s_i,t_i)-path for $1\le i\le k-1$.

2. $\mathrm{OPT}_k\cup\mathrm{OPT}_{k-1}=\mathrm{OPT}'_k\cup\mathrm{OPT}'_{k-1}$.

The first property guarantee that OPT'_k and OPT'_{k-1} are feasible solutions to $\mathrm{SDPP}(\{(s_1, t_1), \ldots, (s_{k-1}, t_{k-1})\})$ and $\mathrm{SDPP}(\{(s_1, t_1), \ldots, (s_k, t_k)\})$, respectively. Property (2) implies that OPT'_k and OPT'_{k-1} are optimal, thus, by the Uniqueness Assumption, we have $\mathrm{OPT}'_k = \mathrm{OPT}_k$ and $\mathrm{OPT}'_{k-1} = \mathrm{OPT}_{k-1}$.

3.0.1 Untwisting chains of bigons

Let $\Gamma = \mathrm{OPT}_{k-1}$ and $\Pi = \mathrm{OPT}_k$. A *free bigon* B is a connected component of $\mathbb{R}^2 \backslash (\Gamma \cup \Pi)$ whose boundary is composed of exactly one subpath α of Γ and exactly one subpath β of Π. We refer to α and β as *sides* of B, and to their common end vertices as *endpoints* of B.

We say that a simple subpath α of a path of Γ and a simple subpath β of a path of Π *touch* at a vertex or path of $\alpha \cap \beta$ (an *intersection* of α and β) if α and β do not cross.

We say that an endpoint e of B is a touching point if simple superpaths of α and β, of which e is an internal vertex, touch at e. We say that bigon B is (1) *crossed* if neither endpoint of B is a touching point, (2) *uncrossed* if both endpoints of B are touching points and (3) *mixed* otherwise (See Figure 3.1).

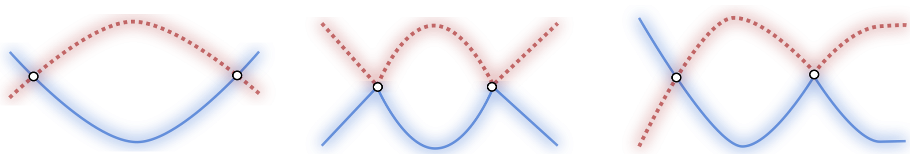

Figure 3.1: (left) crossed bigons (middle) uncrossed bigons (right) mixed

.

We say that the bigon B_1 with sides α_1 and β_1 is inside bigon B_2 with sides α_2 and β_2, if the topological disk bounded by α_1 and β_1 is a subset of the topological disk bounded by α_2 and β_2 in the plane graph.

For the optimal path $\pi_k \subseteq \Pi = \mathrm{OPT}_k$, we define its bigon $\tilde{B}_k$ to be the region bounded by π_k and the s_k-to-t_k path along the outer cycle. Since there is no path of OPT_{k-1} bounding B_k, we will pretend the s_k-to-t_k path along the outer cycle to be in OPT_{k-1} so that from now on, we can treat B_k like other bigons.

Let $C = (B_1, B_2, \ldots, B_t)$ be a sequence of empty bigons and let α_i and β_i be the sides of B_i. We call C a *chain* (of empty bigons) if $\alpha_1 \circ u_1 \circ \alpha_2 \circ u_2 \ldots \circ u_{t-1} \circ \alpha_t$ is a subpath in Γ, $\beta_1 \circ u_1 \circ \beta_2 \circ u_2 \ldots \circ u_{t-1} \circ \beta_t$ is a subpath in Π and u_i $(1 \leq i \leq t-1)$ is a vertices of $\Gamma \cap \Pi$.

Chain C is *straight* if all of its internal bigons are uncrossed and is *twisted* otherwise; chain C is a *chain block* if it is maximally straight. We refer to the $u_1, u_2, \ldots, u_{t-1}$ as the vertices of the chain. A vertex of C is *internal* if it is the common vertex of two consecutive bigons, and it is an *endpoint* otherwise. We say that a chain block is *crossed* if at least one endpoint of the chain block is a crossing point. We say that chain block B is *not free* if there is another chain block $B' \neq B$ that cross B.

Untwist. Let $B = (B_1, B_2, \ldots, B_t)$ be a crossed chain block, and let $\alpha_i \in \Gamma$ and $\beta_i \in \Pi$ be the sides of B_i (for all $1 \leq i \leq t$). Let $\alpha = \alpha_1 \circ \ldots \circ \alpha_t \in \gamma_p \subseteq \Gamma$ (for a $1 \leq p \leq k-1$), and let $\beta = \beta_1 \circ \ldots \circ \beta_t \in \pi_q \subseteq \Pi$ (for a $1 \leq q \leq k$).

The *untwist operation* on the crossed chain block B changes γ_p to $\gamma_p \oplus \alpha \oplus \beta$ and π_q to $\pi_q \oplus \alpha \oplus \beta$ (see Figure 3.2).

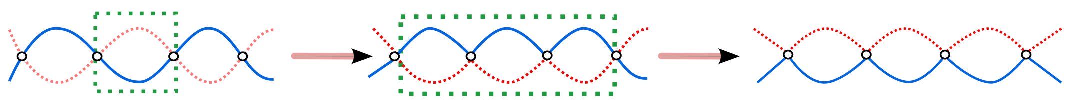

Figure 3.2: A sequence of two bigon-chain twists. The dashed box highlights a chain of bigons that, if we twist them, reduces the number of crossing paths between the red (dashed) and blue (solid) paths.

The following lemma is immediate from the definition of untwisting.

Lemma 3.0.2. *Let B be a crossed chain block with boundaries $\alpha \in \gamma_p$ and $\beta \in \pi_q$. Untwisting B*

(1) preserves $\Gamma \cup \Pi$, and

(2) reduces the total number of crossings.

Proof. To prove (1), notice that the untwist operation on crossed chain block B changes subpath γ_p to $\gamma_p \oplus \alpha \oplus \beta$ and π_q to $\pi_q \oplus \alpha \oplus \beta$. Thus, $\gamma_p - \alpha + \beta$ and $\pi_q - \beta + \alpha$ are the subpaths in Γ and Π after untwisting. Since $(\gamma_p - \alpha + \beta) \cup (\pi_q - \beta + \alpha) = \gamma_p + \pi_q \subseteq \Gamma \cup \Pi$ (for $1 \leq p \leq k-1$ and $1 \leq q \leq k$), the untwisting operation preserves $\Gamma \cup \Pi$.
To prove (2), assume that $B = (B_1, B_2, \ldots, B_t)$ is a crossed chain block. By definition,

at least one of the endpoints of B is a crossing point. Without loss of generality assume that $\alpha_t \oplus u_t \oplus \alpha_{t+1}$ and $\beta_t \oplus u_t \oplus \beta_{t+1}$ cross at u_t. Untwisting the crossed chain block B will reduce the number of crossings by 1. □

The following lemma is implied by Hass and Scott [13], lemma 3.1. In their paper they have proved the existence of a free bigon.

Lemma 3.0.3. *If a path of Γ crosses a path of Π, then there is a free crossed chain block.*

Proof. Let $B = (B_1, B_2, \ldots, B_t)$ be a crossed chain of bigons. If B is free we are done. Otherwise, there exists a subpath of π_q (for a $1 \leq q \leq k$) that has crossed a subpath of γ_p (for a $1 \leq p \leq k-1$). Let t_1, t_2 be the crossing points of the π_q with α respectively. Let $B' = \alpha[t_1, t_2] \oplus \pi_q[t_1, t_2]$ be our new bigon. Since B' is inside B, doing the same operation inside B' recursively will reduce the problem size and finally we get a *free* bigon. □

Lemma 3.0.2 and Lemma 3.0.3 imply that we can cancel all crossings by repeatedly untwisting crossed chain blocks, see Figure 3.3

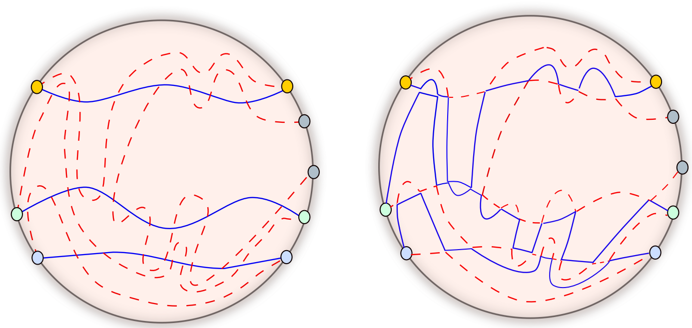

Figure 3.3: (left)The optimal solution $\{Q_1, Q_2, \ldots, Q_{k-1}\}$ are colored blue and the optimal solution $\{\bar{Q}_1, \bar{Q}_2, \ldots, \bar{Q}_k\}$ are colored red. (right) The result after switching bigon to eliminate crossings.

Lemma 3.0.4. *Let $\Gamma = \mathrm{OPT}_{k-1}$ and $\Pi = \mathrm{OPT}_k$. There exists $\Gamma' = (\gamma'_1, \ldots, \gamma'_{k-1})$ and $\Pi' = (\pi'_1, \ldots, \pi'_k)$ with the following properties.*

1. $\Gamma \cup \Pi = \Gamma' \cup \Pi'$.

2. No pair of paths in $\Gamma' \cup \Pi'$ *cross.*

Proof. Let $C = (B_1, B_2, \ldots, B_t)$ be a chain of bigons. By Lemma 3.0.3 there exist a *free* crossed chain block in $C \subseteq \Gamma \cup \Pi$. By Lemma 3.0.2 untwisting it will reduce the number of crossings by 1 and will preserve $\Gamma \cup \Pi$. Since the number of crossing points is a non-negative number and each untwisting operation reduces the number of crossing points by one, by repeatedly doing the untwisting operation we will get rid of all crossing points. □

We emphasize that the paths in Γ' are not necessarily disjoint. Similarly, the paths in Π' are not necessarily disjoint. However, there exists no crossing points between Γ' and Π'. Next, we need to twist some chains back to obtain feasible solutions at the price of introducing a bounded number of crossing points.

3.0.2 Twisting back.

Since there is no crossing point in the arrangement of Γ' and Π', all bigons are uncrossed. Let B be a bigon in the arrangement of Γ' and Π', with sides α and β. We say that B is *internal* if (1) $\alpha \in \gamma'_i$ and $\beta \in \pi'_i$ for the same index i, and (2) B is inside the weakly simple polygon with boundary $\gamma'_i \circ \pi'_i$. The *index* of the bigon B is identical to indices of its sides in Γ and Π.

Lemma 3.0.5. *Let* $\{B_1, B_2, \ldots, B_t\}$ *be the set of all internal bigons in the arrangement of* Γ' *and* Π'*, with sides* $\{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_t, \beta_t)\}$*. The set* $\mathcal{P}_{\Gamma'} = \{\alpha_1, \alpha_2, \ldots, \alpha_t\}$ *is a partition of the edge set* Γ'*, and the set* $\mathcal{P}_{\Pi' \backslash \pi'_k} = \{\beta_1, \beta_2, \ldots, \beta_t\}$ *is a partition of the edge set* $\Pi' \backslash \pi'_k$*.*

Proof. To prove that $\mathcal{P}_{\Gamma'} = \{\alpha_1, \alpha_2, \ldots, \alpha_t\}$ is a partition of Γ' we should show that (1) if $X \in \mathcal{P}_{\Gamma'}$, then X is not empty (2) If $X \neq Y$ are elements from $\mathcal{P}_{\Gamma'}$, then $X \cap Y = \emptyset$, and (3) $\cup_{i=1}^{t} \alpha_i = \Gamma'$. To prove (1), let $X \in \mathcal{P}_{\Gamma'}$ be an arbitrary element, then by definition, each subpath α_i of B_i (for a $1 \leq i \leq t$) is non-empty. To prove (2), let $X, Y (X \neq Y)$ be two members of $\mathcal{P}_{\Gamma'}$, then by definition, since Γ' is a set of vertex-disjoint paths, no two member share an edge. Hence, $X \cap Y = \emptyset$. To prove (3), Let $e \in \Gamma'$ be an arbitrary edge. Since Γ' is a set of vertex-disjoint paths no two duplicate edge exist in Γ'. Moreover, by Lemma 3.0.4 we will not remove any edge from $\Gamma' \cup \Pi'$. □

We color all paths in Γ' red and all paths in Π' blue, and refer to them by their color. For example, each internal bigon has exactly one red side and one blue side. If no two sides of same color touch each other then Γ' and Π' are already feasible solutions and we are done. Otherwise, we need to twist internal chains to obtain feasible solutions Γ'' and Π''. Our operations introduce at most $O(k^2)$ number of crossing points only between paths with identical indices.

Lemma 3.0.6. *Each vertex is incident to at most 4 edges of* $\Gamma' \cup \Pi'$

Proof. Let v be an arbitary vertex in $\Gamma' \cup \Pi'$. Now look at the vertex v in our initial system $\Gamma \cup \Pi$. Since $\Gamma \cup \Pi$ is a set of vertex-disjoint paths, at most two subpaths of Π are incident to v. Similarly, at most two subpaths of Γ are incident to v, so the degree of vertex v in $\Gamma \cup \Pi$ is at most 4. Moreover, by Lemma 3.0.4, since $\Gamma \cup \Pi = \Gamma' \cup \Pi'$, the set of edges incident to vertex v is fixed, which completes the proof. □

3.0.2.1 Kissing Graph

We define the *kissing* (muti-)graph $H = (I, K)$ as follows. The vertex set, I is composed of one vertex b_i for each internal bigon. For each two vertices $b, b' \in I$ that have a common touching vertex, we add an edge between them. We label each edge with its corresponding touching vertex (See Figure 3.4).

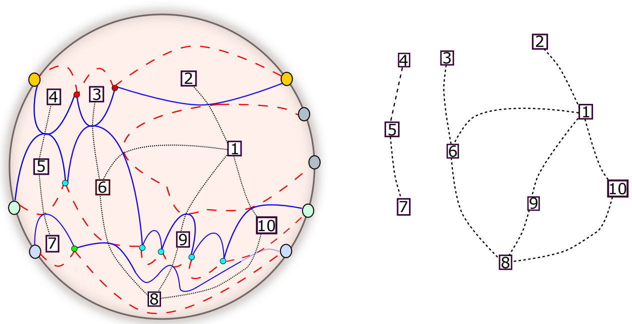

Figure 3.4: *Kissing Graph* H of the graph G. The vertex set I is composed of one vertex b_i for each internal bigon.

Let $W = (b_1, b_2, \ldots, b_t)$ be a walk in the kissing graph H. The bigon $b_i = B_i$ *crosses* W if (b_{i-1}, b_i) and (b_i, b_{i+1}) correspond to touching vertices on different sides of B_i. The

bigon B_i is *tangent* to W if (b_{i-1}, b_i) and (b_i, b_{i+1}) correspond to touching vertices on the same side of B_i.
We have the following observation:

Observation 3.0.7. *Let v be a touching point and let e_1, e_2, e_3, e_4 be edges in $\Gamma \cup \Pi = \Gamma' \cup \Pi'$ that are incident to v in clockwise order, then if $e_1, e_2 \in \Pi$ or $e_1, e_2 \in \Gamma$, then $e_1, e_2 \in \Pi'$ or $e_1, e_2 \in \Gamma'$.*

Lemma 3.0.8. *Let C be a cycle in H, let t be the total number of bigons tangent to C. Then, t is an even number.*

Proof. First, we show the lemma for the case that C is the boundary of a face of H. Suppose for the sake of contradiction that an odd number of bigons are tangent to C. We build a cycle C'; for each *Tangent* bigon we add one segment in the cycle C' and for each *Crossed* bigon we add two segments. Let $v_1, v_2, \ldots, v_{2k+1}$ be the touching points of these segments. Let $u_1, u_2, \ldots, u_{2k+1}$ be the corresponding vertices of $v_1, v_2, \ldots, v_{2k+1}$ in $\Gamma \cup \Pi$, respectively. Then, since C' is composed of an odd number of segments, there exist $i, 1 \leq i \leq 2k+1$ such that two segments $u_{i-1}u_i$ and u_iu_{i+1} have the same color. Thus, they are two sub-paths sharing u_i of the same path. Hence, u_i must be a touching point. However, by Observation 3.0.7, the order of the segments along a touching vertex u_i is fixed.

Now, we show the lemma holds for any cycle C inductively. If C is not the boundary of a face then there is a sub-chain γ that is inside C and that connects two bigons x and y on the boundary of C. Let F_1 and F_2 be the two cycles inside C that are divided by γ. Let a, b, c be the number of tangent bigons on $\gamma \backslash \{x, y\}$, $F_1 \backslash \gamma$, and $F_2 \backslash \gamma$, respectively. There are two cases to consider for each x and y:

- **Case I**: x/y is a bigon with three touching points on one of it sides.

- **Case II**: x/y is a bigon with two touching points on one of its sides and one touching point on the other side

This results in a four different combinations for x and y. We know that $a + c + [x \text{ tangent to } F_1] + [y \text{ tangent to } F_1]$, and $b + c + [x \text{ tangent to } F_2] + [y \text{ tangent to } F_2]$ have an even parity. Now we will show that C has also an even parity. To this end, we should consider all possible combinations for x and y. Here, we will go through one combination; all other combinations are straightforward to verify. If x and y are both *Case I* bigon, then $a+b+[x \text{ tangent to } C]+[y \text{ tangent to } C] = a+b+1+1 = a+b+2$ which is an even number since F_1 and F_2 have an even number of tangent bigons we know that $a+c+b+c = a+b+2c$ is an even number which in term $a+b+2$ is an even. To prove the other parts we similarly iterate through possible values that x and y can take and compare it to the number of tangent bigons in F_1 and F_2. $\square$

3.0.2.2 Look at the spanning tree to twist chains.

We re-color the sides of the internal bigons to obtain two feasible solutions Γ'' and Π'' the red and the blue solution, respectively. Let T be a spanning tree of H rooted at R. Fix the color of π'_k to be blue.

Let B be any internal bigon (of H), with sides α and β. Let B' be the parent of B, let $e = (B, B') \in T$, and suppose e intersects α (and not β). We call α the *entrance* of B on T. We color α red and β blue if an even number intermediate vertices of $T[r, v]$ are tangent to this tree path. Otherwise, we color α blue and β red.

Lemma 3.0.5 implies that each edge of $\Gamma' \cup \Pi'$ is colored either blue or red. Let the collection of red edges be Γ'' and the collection of blue edges be Π''.

Lemma 3.0.9. *We have:*

1. Γ'' *is a set of disjoint paths between* $\{(s_1, t_1), \ldots, (s_k, t_k)\}$.
2. Π'' *is a set of disjoint paths between* $\{(s_1, t_1), \ldots, (s_{k-1}, t_{k-1})\}$.
3. *For each* $\gamma''_i \in \Gamma''$ *and* $\pi''_j \in \Pi''$ *such that* $i \neq j$, γ''_i *and* π''_j *do not cross.*
4. $\Gamma'' \cup \Pi'' = \Gamma' \cup \Pi'$.

Proof. To prove (1,2): Let $e \in K$ be a touching vertex between the bigons B and B'. Let α and β be the sides of B, and let α' and β' be the sides of B'. Suppose $e \in \alpha \cap \alpha'$. We prove that α and α' have different colors. By Lemma 3.0.8 the parity of tangent bigons is even in any cycle in H. Moreover, the spanning tree structure guarantees that no two bigons touch each other with the same color.

To prove (3): Since we only consider internal bigons, that have their sides in paths with the same index, our untwists can only introduce crossing points between walks with the

same index.
To prove (4): In all twisting operations we substitute edges between different colors, so their union is invariant.

□

Recall that I is the set of internal bigons. Let Y be a connected region in $\mathbb{R}^2 \backslash I$. We call I a multi-sided region if its boundary intersect at least three different chains, or at least two different chains and π'_k.

Lemma 3.0.10. *There are O(k) multi-sided regions.*

Proof. We define a graph T (that we will argue is a tree) whose vertices are multi-sided regions $\mathcal{M} = (\eta_1, \eta_2, \ldots, \eta_q)$. Two distinct vertices i and j are incident in this graph if there is an endpoint x of η_i and y of η_j that are connected by an x-to-y curve in $\mathbb{R}^2 \setminus \partial G$ that does not cross any bigon and does not go inside any region. Note that this curve may be trivial (i.e. $x = y$). The vertices of T (in a non-degenerate instance) correspond one-to-one with components of $\mathcal{M}$. The edges of T cannot form a cycle, since by the Jordan Curve this would define a disk that is disjoint from ∂G. Therefore T is indeed a tree. Now we will argue that the the number of vertices in this tree is $O(k)$. Note that number of leaf vertices of T could be at most k since each consecutive terminal create at most 1 leaf region. Since the average degree of vertices in a tree is 2 and the degree of each vertex in T is 3 (Because of *multi-sided* region), then the number of edges in the tree with k leaves is $O(k)$. □

Let B and B' be two internal bigons of a chain C in $\Gamma' \cup \Pi'$. We say that B and B' are in conflict if $T[B, B']$ has odd parity (odd number of edges), the total number of tangent bigons to it has the same parity as the number of times it contains R.

An *alternating polygon* Σ is an alternating sequence $(C_0, T_0, C_1, T_1, \ldots, C_{t-1}, T_{t-1}, C_t = C_0)$, where C_i's are sub-chains of internal chains and T_i's are subpaths of T . The last bigon of each C_i is identical to the first bigon of T_i, and the last bigon of T_i is identical to the fist bigon of C_{i+1}.

Lemma 3.0.11. *Let* $\Sigma = (C_0, T_0, C_1, T_1, \ldots, C_{t-1}, T_{t-1}, C_t = C_0)$ *be an alternating polygon. Let* t *be the number of bigons that are tangential to all* T_i*'s and let* t' *be the number of* C_i*'s whose end bigons are in conflict. If* $\iota(\Sigma)$ *does not contain any terminal and* $t + t'$ *is odd then* $\iota(\Sigma)$ *contains a multi-sided region.*

Proof. We use induction on the total number of bigons that cross T_i's.

Base case: No bigon crosses T_i. Hence, all of the bigons are tangent. Since the intersection of tangent bigons will create closed disk, by definition, it will create a *multi-sided* region.

Induction step: Suppose, there is one bigon B that crosses some T_i, and suppose B belongs to a chain C. Since Σ does not contain any terminal, then C must leave Σ, therefore, there must be $B' \in C$ that crosses Σ. $C[B, B']$ splits Σ into two smaller alternating polygons Σ_1 and Σ_2. Let t_1 and t_2 be the total number of tangential bigons to Σ_1 and Σ_2, respectively. Also, let t'_1 and t'_2 be the number of chains with conflicting endpoints in Σ_1 and Σ_2, respectively. Note that $t = t_1 + t_2$. Also, $t' = t'_1 + t'_2$ if B and B' are not in conflict, and $t' = t'_1 + t'_2 - 2$ otherwise. In any case, at least one of $t_1 + t'_1$ or $t_2 + t'_2$ is odd. So, the induction hypothesis implies that at least one of Σ_1 or Σ_2 contains a muti-sided region. Thus, Σ contains a multi-sided region. □

Summing up.

Proof of Theorem 3.0.1. Let $\Gamma = \mathrm{OPT}_{k-1}$ be the shortest set of disjoint paths for $\{(s_1, t_1), \ldots, (s_{k-1}, t_{k-1})\}$. Let $\Pi = \mathrm{OPT}_k$ be the shortest set of disjoint paths for $\{(s_1, t_1), \ldots, (s_k, t_k)\}$. Lemma 3.0.4 and Lemma 3.0.9 imply the existence of Γ'' and Π''. Since $\Gamma \cup \Pi = \Gamma'' \cup \Pi''$, Γ and Π are vertex-disjoint and Γ'' and Π'' are vertex-disjoint, we have $|\Gamma| + |\Pi| = |\Gamma''| + |\Pi|$. Additionally, $|\Gamma| \leq |\Gamma''|$ and $|\Pi| \leq |\Pi''|$ because of the optimality of Γ and Π. Therefore, $|\Gamma| = |\Gamma''|$ and $|\Pi| = |\Pi''|$. The uniqueness of the optimal solutions implies that $\Gamma = \Gamma''$ and $\Pi = \Pi''$. The theorem follows from the properties of Γ'' and Π'' implied by Lemma 3.0.9.

□

Chapter 4: Conclusion

In Chapter 2, we started with defining three types of path routing problems: Vertex-Disjoint Paths, Edge-Disjoint Paths, and Non-Crossing Paths. In this thesis, we have focused on the optimization version of the Vertex-Disjoint Paths problem in planar graphs. In this problem, a graph G and k pairs of terminals that are on the boundary of the outer face of a planar graph is given and the task is to find a set of k disjoint paths connecting s_i to t_i (For $1 \leq i \leq k$). In Chapter 3, we proposed a structural theorem for solving the k Shortest Vertex-Disjoint Paths problem.

4.0.1 Future Work

It is interesting to further investigate whether we could generalize our method for more general cases in planar graphs. It is challenging to see whether there exists a fixed parameter tractable algorithm for the case when all pairs of terminals are on the boundary of a single face. We still do not know whether the restricted case of the problem where all of the terminals are on the boundary of the outer face lies in P or not.

Bibliography

[1] Matthew Andrews and Lisa Zhang. Hardness of the undirected edge-disjoint paths problem. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 276–283. ACM, 2005.

[2] Andreas Björklund and Thore Husfeldt. Shortest two disjoint paths in polynomial time. In *Automata, Languages, and Programming*, pages 211–222. Springer, 2014.

[3] Glencora Borradaile. Exploiting planarity for network flow and connectivity problems. 2008.

[4] Glencora Borradaile, Claire Kenyon-Mathieu, and Philip Klein. A polynomial-time approximation scheme for steiner tree in planar graphs. *SODA*, 7:1285–1294, 2007.

[5] Glencora Borradaile and Philip Klein. An o (n log n) algorithm for maximum st-flow in a directed planar graph. *Journal of the ACM (JACM)*, 56(2):9, 2009.

[6] Glencora Borradaile, Amir Nayyeri, and Farzad Zafarani. Towards single face shortest vertex-disjoint paths in undirected planar graphs. In *Algorithms-ESA 2015*, pages 227–238. Springer, 2015.

[7] Chandra Chekuri and Sanjeev Khanna. Edge disjoint paths revisited. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 628–637. Society for Industrial and Applied Mathematics, 2003.

[8] Chandra Chekuri, Sanjeev Khanna, and F Bruce Shepherd. Edge-disjoint paths in planar graphs with constant congestion. *SIAM Journal on Computing*, 39(1):281–301, 2009.

[9] Marek Cygan, Dániel Marx, and Michal Pilipczuk. The planar directed k-vertex-disjoint paths problem is fixed-parameter tractable. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 197–206. IEEE, 2013.

[10] Jeff Erickson and Amir Nayyeri. Shortest non-crossing walks in the plane. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 297–308. SIAM, 2011.

[11] Steven Fortune, John Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111 – 121, 1980.

[12] Venkatesan Guruswami, Sanjeev Khanna, Rajmohan Rajaraman, Bruce Shepherd, and Mihalis Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *Journal of Computer and System Sciences*, 67(3):473–496, 2003.

[13] Joel Hass and Peter Scott. Intersections of curves on surfaces. *Israel Journal of Mathematics*, 51(1-2):90–120, 1985.

[14] Monika R. Henzinger, Philip Klein, Satish Rao, and Sairam Subramanian. Faster shortest-path algorithms for planar graphs. *J. Comput. Syst. Sci.*, 55(1):3–23, August 1997.

[15] Alon Itai, Yehoshua Perl, and Yossi Shiloach. The complexity of finding maximum disjoint paths with length constraints. *Networks*, 12(3):277–286, 1982.

[16] Richard Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1974.

[17] Jon Michael Kleinberg. *Approximation algorithms for disjoint paths problems.* PhD thesis, Citeseer, 1996.

[18] Yusuke Kobayashi and Christian Sommer. On shortest disjoint paths in planar graphs. *Discrete Optimization*, 7(4):234–245, 2010.

[19] Stavros G Kolliopoulos and Clifford Stein. *Approximating disjoint-path problems using greedy algorithms and packing integer programs.* Springer, 1998.

[20] MR Kramer and Jan van Leeuwen. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. *Advances in Computing Research*, 2:129–146, 1984.

[21] Chung-Lun Li, S Thomas McCormick, and David Simchi-Levi. The complexity of finding two disjoint paths with min-max objective function. *Discrete Applied Mathematics*, 26(1):105–115, 1990.

[22] Amir Nayyeri. *Combinatorial optimization on embedded curves.* PhD thesis, University of Illinois at Urbana-Champaign, 2013.

[23] Richard G Ogier, Vladislav Rutenburg, and Nachum Shacham. Distributed algorithms for computing shortest pairs of disjoint paths. *Information Theory, IEEE Transactions on*, 39(2):443–455, 1993.

[24] Evanthia Papadopoulou. k-pairs non-crossing shortest paths in a simple polygon. In *Algorithms and Computation*, pages 305–314. Springer, 1996.

[25] Neil Robertson and Paul D Seymour. Graph minors. XIII. the disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.

[26] Anand Srinivas and Eytan Modiano. Finding minimum energy disjoint paths in wireless ad-hoc networks. *Wireless Networks*, 11(4):401–417, 2005.

[27] Aravind Srinivasan. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. In *focs*, page 416. IEEE, 1997.

[28] Jun-ya Takahashi, Hitoshi Suzuki, and Takao Nishizeki. Algorithms for finding non-crossing paths with minimum total length in plane graphs. In *Algorithms and Computation*, pages 400–409. Springer, 1992.

[29] Jun-ya Takahashi, Hitoshi Suzuki, and Takao Nishizeki. Finding shortest non-crossing rectilinear paths in plane regions. In *Algorithms and Computation*, pages 98–107. Springer, 1993.

[30] Kasturi Varadarajan and Ganesh Venkataraman. Graph decomposition and a greedy algorithm for edge-disjoint paths. In *Symposium on Discrete Algorithms: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, volume 11, pages 379–380. Citeseer, 2004.

[31] Éric Colin De Verdière and Alexander Schrijver. Shortest vertex-disjoint two-face paths in planar graphs. *ACM Transactions on Algorithms (TALG)*, 7(2):19, 2011.

[32] Bing Yang, Si-Qing Zheng, and Enyue Lu. Finding two disjoint paths in a network with normalized α+-min-sum objective function. In *Algorithms and Computation*, pages 954–963. Springer, 2005.