Shortest Vertex-Disjoint Two-Face Paths in Planar Graphs

Éric Colin de Verdière

École normale supérieure, Paris

and

Alexander Schrijver

Centrum voor Wiskunde en Informatica, Amsterdam

Let G be a directed planar graph of complexity n, each arc having a nonnegative length. Let s and t be two distinct faces of G; let $s_1, \ldots, s_k$ be vertices incident with s; let $t_1, \ldots, t_k$ be vertices incident with t. We give an algorithm to compute k pairwise vertex-disjoint paths connecting the pairs (s_i, t_i) in G, with minimal total length, in $O(kn \log n)$ time.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical algorithms and problems—*Computations on discrete structures*; *routing and layout*; G.2.2 [**Mathematics of Computing**]: Graph Theory—*Graph algorithms; network problems*; *path and circuit problems*

General Terms: Algorithms, Performance, Theory

Additional Key Words and Phrases: Algorithm, disjoint paths, planar graph, shortest path

1. INTRODUCTION

The *vertex-disjoint paths problem* is described as follows: given any (directed or undirected) graph and k pairs $(s_1, t_1), \ldots, (s_k, t_k)$ of vertices, find k pairwise vertex-disjoint paths connecting the pairs (s_i, t_i), if they exist. This problem is well-known also because of its motivation by VLSI-design.

For a fixed number k of pairs of terminals, this problem is polynomial-time solvable in a directed planar graph, as shown by Schrijver [1994], and in any undirected graph, as shown by Robertson and Seymour [1995]. However, Raghavan [1986] and Kramer and van Leeuwen [1984] proved that it is NP-hard when k is not fixed, even on a planar undirected graph; it belongs to the more general class of *integer multicommodity flow problems* [Schrijver 2003, Chapter 70], many variants of which are NP-hard.

If the graph is planar, two special cases are solvable in time linear in the complexity of the graph, even if k is not fixed:
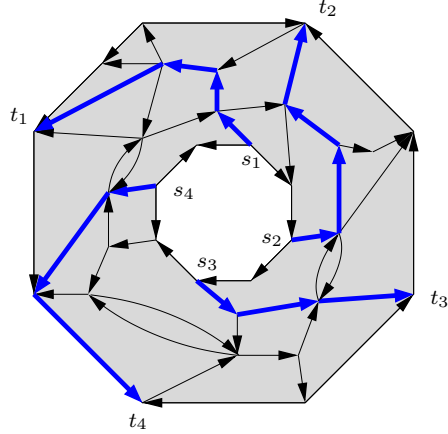
Fig. 1.   An instance of the problem and a solution (in bold lines).

(1) if all terminals lie on the outer face, as proved by Suzuki et al. [1990];
(2) if the terminals $s_1, \ldots, s_k$ are incident with a common face s, the terminals
     $t_1, \ldots, t_k$ are incident with a common face t, and the faces s and t are distinct,
     as proved by Ripphausen-Lipa et al. [1996].

In this paper, we consider a graph where each edge has a nonnegative length, and we wish to solve the vertex-disjoint paths problem using paths with minimal total length. Of course, this is harder than the vertex-disjoint paths problem. In case (1), the problem is known to be solvable in polynomial time (even if k is not fixed) if the cyclic order of the terminals is $s_1, \ldots, s_k, t_k, \ldots, t_1$ (by reduction to the max-flow problem, after replacing each vertex by two vertices connected by an arc, so that the problem is to find arc-disjoint paths in this new graph) [van der Holst and de Pina 2002]. Our goal is to solve the vertex-disjoint paths problem with minimal total length in case (2). We give an algorithm to do this in $O(kn \log n)$ time (see Figure 1):

THEOREM 1.1. *Let G be a planar directed graph with n vertices and arcs, each arc having a nonnegative length. Let s and t be two distinct faces of G; let $s_1, \ldots, s_k$ be vertices incident with s; let $t_1, \ldots, t_k$ be vertices incident with t. Then we can compute k pairwise vertex-disjoint paths connecting the pairs (s_i, t_i) in G, with minimal total length, in $O(kn \log n)$ time.*

The value of k is not fixed in this result. Note that this theorem also holds if G is an undirected graph: simply replace every edge of this graph by two oppositely directed arcs and apply the previous result to this new graph. The same problem for *non-crossing* shortest paths, that is, paths that are allowed to overlap along vertices and edges but not to cross in the plane, is solvable in $O(n \log n)$ time, as shown by Takahashi et al. [1996].

The high-level approach of our algorithm is the following. We first show that we may assume without loss of generality that G satisfies some additional properties and transform G into another planar directed graph D; in this graph, it suffices to solve the same problem for *arc-disjoint* instead of *vertex-disjoint* paths (Section 2).
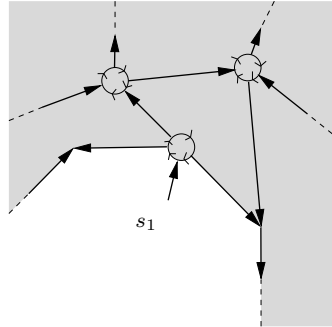
Fig. 2. Construction of the graph $D = (V, A)$ from the graph G. The thin arcs on the rings have length zero.

Then we translate our problem in terms of (integer) flows in the graph D (Section 3). In Section 4, we introduce the residual graph and state some of its properties that we will use. In Section 5, we explain how to increase the value of a flow. By repeated applications of this algorithm, we obtain vertex-disjoint paths in G between the terminals, but they may fail to connect the pairs (s_i, t_i). We show that it suffices to "rotate" the flow a few times to change the connections between the terminals (Section 6) and explain how to do that efficiently (Section 7). A generalization of the notion of potential allows us to assume that all lengths in the residual graph are nonnegative, which makes the algorithm efficient.

2. PRELIMINARIES

We assume that we are given an embedding of the directed graph G in the plane. More precisely, only a *combinatorial embedding* of G is necessary, which means that the cyclic order of the arcs around a vertex is known.

We can assume that G is connected and that t is the outer face of the embedding of G. Up to re-indexing the pairs (s_i, t_i), we may assume that $s_1, \ldots, s_k$ and $t_1, \ldots, t_k$ are in clockwise order: indeed, if such a reordering does not exist, then there cannot exist vertex-disjoint paths connecting the pairs (s_i, t_i).

We may assume that each terminal vertex has degree one as follows: to each terminal vertex s_i (resp. t_i), attach an arc (of length zero, for example) (s_i', s_i) (resp. (t_i, t_i')) inside s (resp. t), where s_i' (resp. t_i') is a new vertex; use the s_i' and the t_i' as terminals, instead of the s_i and the t_i. Clearly, any solution to the problem in this augmented graph yields a solution in the original graph G.

We now transform G into a graph D of maximal degree three, and prove that solving the shortest vertex-disjoint paths problem in D gives a solution of the same problem in G. The graph D is obtained from G by replacing each vertex v of degree at least four by a small clockwise "ring" of arcs; see Figure 2. In particular, the number of vertices and arcs of D is $O(n)$. Every arc a of D that is on no ring corresponds to an arc of G and its length, $\lambda(a)$, is the length of this arc in G; it is thus nonnegative. The length $\lambda(a)$ of an arc a on a ring is zero. The function λ is fixed in this whole paper.

An (s, t)-*path* in D or G is a path from some vertex in $\{s_1, \ldots, s_k\}$ to some vertex
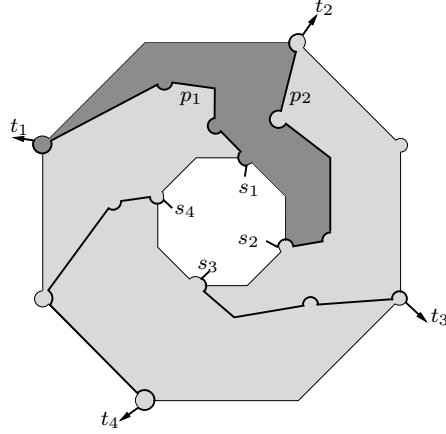
Fig. 3. Illustration of the proof of Proposition 2.1: Every point of the dark shaded area of the annulus is on the right of exactly one path, p_1, and on the left of exactly one path, p_2.

in $\{t_1, \ldots, t_k\}$; an (s_i, t_i)-*path* is a path connecting some pair of terminals (s_i, t_i).

PROPOSITION 2.1. *Let P be a minimum-length set of k vertex-disjoint (s_i, t_i)-paths in D. Then P gives, in $O(n)$ time, a minimum-length set of k vertex-disjoint (s_i, t_i)-paths in G. If no such set P exists, then the original problem in G has no solution.*

PROOF. Consider such a set of (s_i, t_i)-paths P in D. We claim that a given ring r of D can be used by at most one path in P. Indeed, since s and t are distinct faces, $\mathbf{R}^2 \setminus \{s \cup t\}$ is an annulus. Since the paths in P are vertex-disjoint and connect s to t, every point of the annulus that does not belong to a path in P is on the left of exactly one path and on the right of exactly one path in P (Figure 3). In particular, the center c of r is on the right of exactly one path in P. But every path using r has c on its right, because the arcs of r are oriented clockwise. This proves the claim.

Thus, P corresponds, in G, to k pairwise vertex-disjoint (s_i, t_i)-walks. Removing the loops from these walks in $O(n)$ time does not increase the total length and gives a set of k vertex-disjoint (s_i, t_i)-paths in G.

Conversely, any solution of the original vertex-disjoint problem in G gives a set of k vertex-disjoint paths in D, of the same length, connecting the appropriate pairs of terminals. So the paths obtained in the previous paragraph have minimal total length; furthermore, if no such set of paths P exists, then the problem in G admits no solution. □

So we reduced the problem in G to the same problem in the graph D. The point now is that the vertices of D have degree three, except the terminals, which have degree one; because of these degree conditions, a set of *arc*-disjoint (s, t)-paths or circuits in D is actually a set of *vertex*-disjoint (s, t)-paths or circuits in D, so we now have to solve a problem on *arc*-disjoint paths. This enables a flow approach on D, which we will develop in the next section.
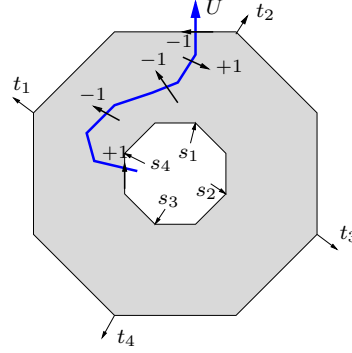
Fig. 4. The path U in the dual graph D^* and the corresponding value of u on the arcs of D. Only the non-zero values of u are indicated, on the arcs in bold lines. Here $m = m_s - m_t = 3 - 1 = 2$.

3. FLOWS AND WINDING NUMBERS

In this paper, a *flow in $D = (V, A)$* is an element $x \in \{0, 1\}^A$ such that, for each non-terminal vertex v, the following *flow conservation law* holds:

$$\sum_{a \mid v = \text{source}(a)} x(a) = \sum_{a \mid v = \text{target}(a)} x(a).$$

The *value* of a flow x equals the total flow leaving the vertices $s_1, \ldots, s_k$: if a_i is the arc incident with s_i, then the value of x equals $\sum_{i=1}^{k} x(a_i)$. A *circulation* is a flow of value zero. A *length function* (or *cost function*) on D is an element of $\mathbf{R}^A$; λ is a length function. The *length* (or *cost*) of a flow x with respect to a length function κ is $\kappa^\top x$.

Note that our definition requires flows in D to be in $\{0, 1\}^A$. Hence, equivalently, a flow is a set of arc-disjoint (s, t)-paths and circuits in D. Actually, by the degree conditions on D, it is a set of *vertex-disjoint* (s, t)-paths and circuits.

Let A^{-1} be the set of arcs in A with reverse orientation. If $\kappa \in \mathbf{R}^A$ is a length function, we define the length of an arc $a^{-1} \in A^{-1}$ to be $\kappa(a^{-1}) = -\kappa(a)$.

Let $X \in \mathbf{R}^{A \cup A^{-1}}$; we define $z^X \in \mathbf{R}^A$ by $z^X(a) = X(a) - X(a^{-1})$. If γ is a walk in $(V, A \cup A^{-1})$, by a slight abuse of notation, we define z^γ to be z^X, where $X(a)$ (resp. $X(a^{-1})$) is the number of times γ travels through the arc a (resp. a^{-1}). The length of γ with respect to a length function κ is thus $\kappa^\top z^\gamma$.

We now want to take into account how a flow "turns around" the inner face s of G. This is done by recording the algebraic amount of flow crossing a fixed path U connecting faces s and t. In more details, consider the (undirected) dual graph D^* of D, that is, the planar graph that has one vertex f^* inside each face f of D and such that f_1^* and f_2^* are connected by an edge e^* if and only if f_1 and f_2 are separated by an arc e in D; in that case, e^* crosses e but no other arc of D. Let U be a path (fixed in this whole paper) from s^* to t^* in D^* (Figure 4). For each arc a in A, define $u(a)$ to be 0 if a does not cross U, $+1$ if a crosses U from left to right, and -1 if a crosses U from right to left. This defines an element $u \in \mathbf{R}^A$. The *winding number* of a flow x equals $u^\top x$, the value of the flow through u counted algebraically. Also, for any $X \in \mathbf{R}^{A \cup A^{-1}}$, the winding number of X is $u^\top z^X$.

Let $m_s \in [1, k]$ be such that the first arc of U is, in the cyclic order around the face s, between s_{m_s} and s_{m_s+1} (where $s_{k+1} = s_1$). Similarly, let m_t be such that the last arc of U is between t_{m_t} and t_{m_t+1}. Let $m = m_s - m_t$.

The following lemma will be used repeatedly.

LEMMA 3.1. *Let γ be any circuit in $(V, A \cup A^{-1})$. Then the winding number of γ belongs to $\{-1, 0, +1\}$. If γ encloses s in the plane, then it has winding number $+1$ if it is clockwise and -1 if it is counter-clockwise. Otherwise, γ has winding number 0.*

PROOF. This is a consequence of the Jordan curve theorem. The winding number of γ is the number of times the path U crosses γ from the right to the left, minus the number of times U crosses γ from the left to the right. Assume γ is clockwise, the other case being analogous. The winding number of γ is the number of times U exits the region enclosed by γ minus the number of times it enters this region.

If γ does not enclose s, then both endpoints of U are outside γ, so the winding number is zero. If γ encloses s, the source of U is inside the region enclosed by γ while its target is outside, so the winding number is $+1$.   □

We can now reformulate our arc-disjoint paths problem in D in terms of flows in D:

PROPOSITION 3.2. *Let x be a flow in D of value k with minimal cost subject to the condition that its winding number, modulo k, equals m. Then x gives, in $O(n)$ time, k vertex-disjoint (s_i, t_i)-paths in D of minimal total length. If there exists no such flow, then there does not exist k vertex-disjoint (s_i, t_i)-paths in D.*

PROOF. As noted above, the degree conditions on D imply that the flow x is a set of *vertex*-disjoint (s, t)-paths or circuits in D. Let γ be a circuit in x. If γ has non-zero winding number, then γ separates s and t, which implies that x has value zero, a contradiction. If γ has winding number zero, then removing it from x yields another flow with the same properties. Since we can remove such circuits in $O(n)$ time, we may assume that x contains only (s, t)-paths. By the assumption on the winding number, these paths connect the pairs (s_i, t_i), for $i = 1, \ldots, k$.

Furthermore, any k vertex-disjoint (s_i, t_i)-paths in D correspond to a flow in D of value k and of winding number equal, modulo k, to m. It follows that the paths obtained have minimal total length.   □

By Propositions 2.1 and 3.2, to prove Theorem 1.1, it suffices to show that we can, in $O(kn \log n)$ time, find a flow in D of value k and with minimal cost subject to the condition that its winding number, modulo k, equals m. This will be done in two steps. We will first compute a minimum-cost flow in D of value k, without any requirement on its winding number; this can be done by k flow augmentations along a path (Section 5). It turns out that the winding numbers of this flow and of the solution flow differ by less than k (Section 6); it thus suffices to perform at most k "rotations" of the flow (changing its winding number by one) to get the solution (Section 7).

4. THE RESIDUAL GRAPH

In this section, we introduce the *residual graph* of D; it is a classical tool for dealing with maximal flows and flows of minimal cost [Schrijver 2003, Chapters 10–12].

Let x be a flow on $D = (V, A)$. Let A_x be the subset of $A \cup A^{-1}$ defined by

$$A_x = \{a \mid x(a) = 0\} \cup \{a^{-1} \mid x(a) = 1\}.$$

The *residual graph* of D with respect to x is the directed graph $D_x = (V, A_x)$; it is thus the graph obtained from D by reversing the sign of the length and winding number and the orientation of the arcs a such that $x(a) = 1$.

The following lemma explains the interest of the residual graph; the first two assertions are well-known.

LEMMA 4.1. *Let x be a flow in D.*

(1) *D_x has no (s, t)-path if and only if x has maximal value in D among all flows.*

(2) *Assume that x has maximal value in D; let κ be a length function. Then D_x has no negative-length directed circuit with respect to κ if and only if x has minimal cost, with respect to κ, among all flows in D with the same value.*

(3) *Assume x has maximal value in D. Then D_x has no directed circuit with winding number one if and only if x has maximal winding number among all flows in D with the same value.*

PROOF. In these three assertions, the "if" part is easy: If D_x has an (s, t)-path or circuit γ, then, by construction of D_x, $y := x + z^\gamma$ is a flow in D; its cost equals the cost of x in D plus the cost of γ in D_x; its winding number equals the winding number of x plus the winding number of γ; and its value equals the value of x plus one if γ is a path, or the value of x if γ is a circuit.

Conversely, let x and y be flows in D. Consider $y - x$ in the graph D. By construction of D_x, this is a flow in D_x, in the sense that the flow conservation law holds at each vertex of D (except at the terminals) and that, for each arc $a \in A$, we have $(y - x)(a) \geq 0$ if $a \in A_x$ and $(y - x)(a) \leq 0$ if $a^{-1} \in A_x$. In particular, $y - x$ can be written as $\sum_{\gamma \in Z} z^\gamma$, where Z is a set of (s, t)-paths, (t, s)-paths, and circuits in D_x.

Now, to prove the "only if" part of (1), simply note that, if D_x has no (s, t)-path, then there is no (s, t)-path in Z; thus, the value of y cannot be greater than the value of x. To prove the "only if" part of (2) and (3), assume that x and y both have maximal value in D. Then, by (1), Z contains no (s, t)-path, hence also no (t, s)-path, hence only circuits. If D_x has no negative-length directed circuit, the cost of y is at least the cost of x; this proves (2). If D_x has no directed circuit with winding number one, then y cannot have winding number higher than x, for otherwise $y - x$ would contain at least one circuit with positive winding number, hence with winding number one (Lemma 3.1). This proves (3). □

A length function κ is *nonnegative on D_x* if κ is nonnegative on every arc in A_x; that is, for each $a \in A$, $\kappa(a) \geq 0$ if $x(a) = 0$ and $\kappa(a) \leq 0$ if $x(a) = 1$.

5. INCREASING THE FLOW IN D

In this section, we explain how to compute a minimum-cost flow in D in $O(kn \log n)$ time. The algorithm uses only very classical minimum-cost flow techniques, but we

indicate it for completeness and because Section 7 will use some similar ideas.

Let $p \in \mathbf{Z}$. A *p-flow* is a flow in D of value p. Let κ and κ' be two length functions on D; we write $\kappa \simeq \kappa'$ if $\kappa^\top z^\gamma = \kappa'^\top z^\gamma$ for each closed walk γ in $(V, A \cup A^{-1})$. (This notion is equivalent to the notion of potential.)

LEMMA 5.1. *Let $\kappa \simeq \kappa'$. Then any minimum-cost k-flow with respect to κ is also a minimum-cost k-flow with respect to κ'.*

PROOF. By Lemma 4.1(2), a k-flow x has minimum cost with respect to κ if and only if D_x has no negative-length circuit with respect to κ. Since $\kappa \simeq \kappa'$, circuits in D_x have the same length with respect to κ and to κ'.   □

The following result follows from classical minimum-cost flow techniques.

LEMMA 5.2. *Let x be a p-flow in D and let κ be a length function that is non-negative on D_x. Then, in $O(n \log n)$ time, we can find a $(p+1)$-flow x' and a length function $\kappa' \simeq \kappa$ that is nonnegative on $D_{x'}$, unless x has maximal value.*

PROOF. We temporarily add to D_x two vertices s and t, and arcs (s, s_i) and (t_i, t) of length zero, for $i = 1, \ldots, k$. Let D'_x be the resulting graph. We compute a shortest path tree of D'_x with root s, with respect to κ, in $O(n \log n)$ time using Dijkstra's algorithm [Dijkstra 1959] speeded up with Fibonacci heaps [Fredman and Tarjan 1987], because all lengths are nonnegative[1]. If there is no path from s to t in D'_x, then D_x has no (s, t)-path, hence, by Lemma 4.1(1), x has maximal value.

Otherwise, for each vertex v of D'_x, let $d(v)$ be the distance from s to v with respect to κ, as computed by Dijkstra's algorithm above. For each arc $a = (u, v)$ of A_x, we have $d(v) \leq d(u) + \kappa(a)$ by the triangle inequality, with equality if a is on the shortest path tree. For each arc $a = (u, v)$ of A_x, let $\kappa'(a) = \kappa(a) + d(u) - d(v)$; clearly, $\kappa' \simeq \kappa$. We have $\kappa'(a) \geq 0$, and $\kappa'(a) = 0$ if a is on the shortest path tree. Let γ be the (s, t)-path in D_x corresponding to the path from s to t in D'_x in the shortest path tree. Now, let $x' = x + z^\gamma$; this is a $(p + 1)$-flow in D because γ is an (s, t)-path in D_x. Furthermore, since κ' is nonnegative on the arcs of D_x and is zero on the arcs of γ, it is nonnegative on $D_{x'}$.   □

Starting with the zero flow x (for which $D_x = D$) and the length function $\kappa = \lambda$, we repeatedly apply Lemma 5.2. We obtain a flow x_0 with maximal value p and a length function $\kappa_0 \simeq \lambda$ such that κ_0 is nonnegative on D_{x_0}. This takes $O(pn \log n) = O(kn \log n)$ time. If $p < k$, then the original problem has no solution, hence we stop here. Otherwise, x_0 is a minimum-cost k-flow with respect to κ_0 (since κ_0 is nonnegative on D_{x_0} and by Lemma 4.1(1)), and also to λ (Lemma 5.1).

Let w_0 be the winding number of x_0. If $w_0 \equiv m \pmod{k}$, then we are done (Propositions 2.1 and 3.2); so we henceforth assume $w_0 \not\equiv m \pmod{k}$.

6. FINDING THE WINDING NUMBER

A *(k, w)-flow* is a flow in D of value k and winding number w. Let w_1 and w_2 be the integers equal, modulo k, to m that are the closest to w_0 and satisfy $w_1 < w_0 < w_2$. The following proposition states that the problem boils down to finding minimum-cost (k, w)-flows, for $w = w_1$ and $w = w_2$:

---

[1]We could do that in $O(n)$ time using the algorithm by Henzinger et al. [1997], but that would not change the asymptotic complexity of the entire algorithm.

PROPOSITION 6.1. *There is a minimum-cost flow in D (with respect to λ) of value k and winding number equal, modulo k, to m that is either a (k, w_1)-flow or a (k, w_2)-flow.*

PROOF. For every integer w, let μ_w be the minimal cost of the (k, w)-flows. (It is infinite if no (k, w)-flow exists.) By Lemma 4.1(3), the set $\{w \mid \mu_w < \infty\}$ is an interval of integers.

We show that for every integer w such that μ_{w-1}, μ_w, and μ_{w+1} are finite, we have

$$2\mu_w \leq \mu_{w-1} + \mu_{w+1}. \tag{1}$$

Indeed, let x and x' be minimum-cost $(k, w-1)$- and $(k, w+1)$-flows, respectively. Then $x' - x$ gives a nonnegative circulation in D_x (in the sense that it satisfies the flow conservation law and that, for each $a \in A$, $y(a) \geq 0$ if $a \in A_x$ and $y(a) \leq 0$ if $a^{-1} \in A_x$) of winding number 2. So the support of $x' - x$ contains a directed circuit γ in D_x of positive winding number, hence 1. Then $x + z^\gamma$ and $x' - z^\gamma$ are both (k, w)-flows. Thus

$$2\mu_w \leq \lambda^\top(x + z^\gamma) + \lambda^\top(x' - z^\gamma) = \lambda^\top x + \lambda^\top x' = \mu_{w-1} + \mu_{w+1},$$

which proves (1).

So μ_w is monotonically non-increasing for $w \leq w_0$ and monotonically non-decreasing for $w \geq w_0$. Thus Proposition 6.1 holds. □

7. ROTATING THE FLOW IN D

In this section, we explain how to derive, from a minimum-cost (k, w)-flow, a minimum-cost $(k, w+1)$-flow, and we conclude the proof.

Let κ and κ' be two length functions on D; we write $\kappa \sim \kappa'$ if $\kappa^\top z^\gamma = \kappa'^\top z^\gamma$ for each closed walk γ *with winding number zero* in $(V, A \cup A^{-1})$. Clearly, $\kappa \simeq \kappa'$ implies $\kappa \sim \kappa'$.

PROPOSITION 7.1. *Let $\kappa \sim \kappa'$. Then any minimum-cost (k, w)-flow with respect to κ is also a minimum-cost (k, w)-flow with respect to κ'.*

PROOF. Let x and y be two (k, w)-flows in D. Then $y - x$ is a circulation in $(V, A \cup A^{-1})$, i.e., a sum of terms of the form z^γ, where γ is a circuit in $(V, A \cup A^{-1})$. Furthermore, there are as many circuits with winding number $+1$ as with winding number -1 in this sum.

We have $(\kappa' - \kappa)^\top z^\gamma = 0$ for every such circuit with winding number zero. Moreover, if γ has winding number $+1$ and γ' has winding number -1, it follows from the definition of "$\sim$" that $\kappa^\top(z^\gamma + z^{\gamma'}) = \kappa'^\top(z^\gamma + z^{\gamma'})$. We thus have $\kappa^\top(y - x) = \kappa'^\top(y - x)$, implying the result. □

We view D as an undirected planar graph H; s and t are two faces of H. Let H^* be its dual graph. If e is an *oriented* edge of H, then e^* is the dual edge oriented so that e^* crosses e from right to left.

A *cut* of H^* is a set X^* of *oriented* edges of H^* such that any directed path from s^* to t^* uses at least one oriented edge of X^*. The following lemma is inspired by Reif [1983, Propositions 1 and 2]. See Figure 5.
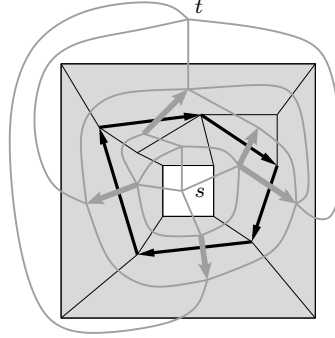
Fig. 5. Illustration of Lemma 7.2: A minimal cut in H^* corresponds to a circuit with winding number one in H. The primal graph H is depicted in black lines, with thicker lines for the arcs of the circuit. The dual graph H^* is depicted in lighter color, with thicker lines for the arcs of the cut.

LEMMA 7.2. *Let X be a set of oriented edges of H. Then X contains the oriented edges of some circuit with winding number one in H if and only if X^* is a cut of H^*.*

PROOF. If we have a directed circuit γ with winding number one, then its dual is a cut. Indeed, consider an (s^*, t^*)-path π in H^*. The face s belongs to the interior of γ, while the face t belongs to the exterior of γ; let e^* be the first oriented edge of π that crosses γ; its source is inside γ while its target is outside γ. By our choice of orientation, e belongs to γ.

Conversely, let X^* be a cut of H^*; we will prove that X contains a circuit with winding number one. Without loss of generality, we may assume that X^* is a cut that is *minimal* with respect to inclusion.

First, label "S" a face f of H if there is, in H^*, a path from s^* to f^* that does not use any oriented edge of X^*. Similarly, label "T" a face f of H if there is, in H^*, a path from f^* to t^* that does not use any oriented edge of X^*. Since X^* is a cut, no face of H is labeled both "S" and "T". We claim that X is precisely the set of oriented edges of H whose right face is labeled "S" and whose left face is labeled "T". Clearly, such edges must belong to X. Conversely, let e be an oriented edge of X; by minimality of X, there is an (s^*, t^*)-path in H^* that avoids $(X \setminus e)^*$ and uses e^* exactly once. Thus the source of e^* is reachable from s^* without using any oriented edge of X^*, and t^* is reachable from the target of e^* without using any oriented edge of X^*. This proves the claim. In particular, every face of H is labeled either "S" or "T".

Let S be the subset of the plane made of the faces labeled "S", together with the *open* edges whose both incident faces are labeled "S". Similarly, let T be the union of the faces labeled "T" together with the open edges whose both incident faces are labeled "T". By the previous paragraph, S and T are disjoint subsets of the plane, and they are connected. Let v be a vertex of H. We claim that there cannot be four faces incident with v, in this cyclic order around v, that belong respectively to S, T, S, and T. This follows from the Jordan curve theorem: assume that we have such faces. Then, by connectivity of S, there is a simple closed curve in $S \cup \{v\}$ that goes through v and has faces of T on both sides of it at v. This curve does

not intersect T and separates T, contradicting its connectivity.

The two previous paragraphs together imply that either X has no edge incident with v, or X has exactly one oriented edge whose target is v and one oriented edge whose source is v. Thus X is a union of vertex-disjoint circuits. Let γ be such a circuit; since S and T are connected, and since the faces on the left (resp. right) of γ are in T (resp. S), γ has winding number one. Hence X contains a circuit with winding number one. $\square$

PROPOSITION 7.3. *Let x be a (k, w)-flow in D and let κ be a length function that is nonnegative on D_x. Then, in $O(n \log n)$ time, we can find a $(k, w+1)$-flow x' and a length function $\kappa' \sim \kappa$ that is nonnegative on $D_{x'}$, unless there is no (k, w')-flow with $w' > w$.*

PROOF. Let e be an oriented edge of H; if e corresponds to an arc a of A_x, then we define the length of e in H to be $\kappa(a) \geq 0$; otherwise, we define the length of e to be ∞. So a walk in D_x corresponds to a walk in H of the same length, and a walk in H corresponds to a walk in D_x if and only if it has finite length. Define the *capacity* $c(e^*)$ of an oriented edge e^* of H^* to be the length of e.

We can detect in $O(n)$ time whether the oriented edges of finite capacity constitute a cut in H^*. If this is not the case, then every cut must use an oriented edge of infinite capacity, hence, by Lemma 7.2, D_x has no circuit of winding number one. It follows that x has maximal winding number among all k-flows, by Lemma 4.1(3). Otherwise, we compute a minimal cut in H^*, which corresponds to a shortest circuit with winding number one in D_x, as follows.

A *flow in H^** is a function φ that associates, to each oriented edge e^* of H^*, a real number that is nonnegative and no greater than $c(e^*)$, such that the flow conservation law holds at each vertex of H^* except at s^* and t^*. The *value* of φ is the total flow leaving s^*.

In $O(n \log n)$ time, we compute a flow φ of maximal value in H^* with respect to these capacities, using the algorithm by Borradaile and Klein [2009]. It is well-known, by the "max-flow min-cut" theorem [Schrijver 2003, Theorem 10.3], that φ corresponds to a cut of minimal cost in H^*: the cut is the set of oriented edges that leave the set of vertices reachable from s^* by using only oriented edges e^* of H^* such that $\varphi(e^*) < c(e^*)$ or $\varphi(e^{*-1}) > 0$.

Such a cut X^* can be computed in $O(n)$ time. Moreover, by replacing all the zero capacities in H^* by infinitesimally small capacities before applying the maximal flow algorithm, we may assume that X^* is a cut that is minimal with respect to inclusion. By Lemma 7.2, we thus obtain a circuit γ of winding number one that has minimal length in D_x.

For each arc a of $A \cup A^{-1}$, let $\kappa'(a) = \kappa(a) - \varphi(a^*) + \varphi(a^{*-1})$; we have $\kappa'(a) = -\kappa'(a^{-1})$, hence this defines a length function. If $a \in A_x$, we have $\varphi(a^*) \leq \kappa(a)$, so $\kappa'(a) \geq 0$. If a belongs to γ, we have $\varphi(a^*) = \kappa(a)$ and $\varphi(a^{*-1}) = 0$, so $\kappa'(a) = 0$.

We claim that $\kappa' \sim \kappa$. By the flow conservation law in H^*, $\kappa' - \kappa$ is a linear combination of functions of the form z^γ, where γ^* is an (s^*, t^*)-path or a circuit in H^*; so it suffices to prove that $z^{\gamma\top}\delta = 0$ for each closed walk δ with winding number zero. But $z^{\gamma\top}\delta$ equals the number of times δ crosses γ^* from left to right minus the number of times δ crosses γ^* from right to left. This always equals zero if γ is a circuit; if γ is an (s^*, t^*)-path, this equals zero because δ has winding

number zero (as in the proof of Lemma 3.1). This proves $\kappa' \sim \kappa$.

Now, let $x' = x + z^\gamma$. The length function κ' is nonnegative on the arcs of D_x and is zero on the arcs of γ, so it is nonnegative on $D_{x'}$. $\square$

To conclude, recall that the k-flow x_0 and the length function κ_0 have been computed in Section 5; $\kappa_0 \sim \lambda$ is nonnegative on D_{x_0}; the integer w_0 is the winding number of x_0 and we have

$$w_0 - k < w_1 < w_0 < w_2 < w_0 + k.$$

Applying iteratively Proposition 7.3, we can find a (k, w_2)-flow x_2 and a length function $\kappa_2 \sim \lambda$ that is nonnegative on D_{x_2}; thus, x_2 is a (k, w_2)-flow of minimal cost with respect to λ, by Lemma 4.1(2) and Proposition 7.1; if no such flow exists, we detect it during the course of the algorithm. Similarly, we can find a minimum-cost (k, w_1)-flow. This takes $O(kn \log n)$ time. By Propositions 6.1, 2.1, and 3.2, the cheapest of these two flows corresponds to the solution. This concludes the proof of Theorem 1.1.

Remark 7.4. Computing winding numbers, with the help of a path U in the dual graph, is actually not necessary for the algorithm. After computing a minimum-cost flow, the only requirement is to rotate the flow clockwise (and then counterclockwise) until we reach a flow with winding number equal, modulo k, to m. But a k-flow satisfies this property if and only if it is an (s_i, t_i)-flow, and that can be tested in $O(n)$ time.

8. CONCLUSION

We have given an algorithm to compute minimum-length vertex-disjoint paths connecting prescribed pairs (s_i, t_i) of terminals in a planar graph, where the s_i and the t_i are incident, respectively, with given faces s and t. The running time is $O(kn \log n)$, where k is the number of pairs of terminals and n is the complexity of the graph.

We note that the techniques developed above allow to solve the same problem, but fixing, in addition, the winding number of the set of paths (or, equivalently, the homotopy classes of the paths in the annulus $\mathbf{R}^2 \setminus \{s \cup t\}$). This can be done by computing a minimum-cost flow in the directed graph D and by rotating the flow until achieving the correct winding number. Since the absolute value of the winding number of a flow is at most n, the complexity of the algorithm is $O(n^2 \log n)$.

Finally, the result of this paper suggests some open questions. How hard is it to solve the minimum-length vertex-disjoint paths in case (1) of the introduction, namely, if all terminals lie on the outer face (not necessarily in the order $s_1, \ldots, s_k, t_k, \ldots, t_1$)? And in the case where all the terminals lie on two faces, but a path may have its two endpoints on the same face? The problem extends to vertex-disjoint *trees* whose leaves are fixed on two faces of the graph (such trees, not necessarily of minimal length, can be computed efficiently [Suzuki et al. 1990]). Also, does our problem remain polynomial-time solvable if each of the terminals has to be incident with one of p prescribed faces of the graph, if p is fixed? What about the same problem for a graph embedded on a surface of fixed genus?

REFERENCES

BORRADAILE, G. AND KLEIN, PH. 2009. An $O(n \log n)$ algorithm for maximum st-flow in a directed planar graph. *Journal of the ACM 56,* 2.

DIJKSTRA, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik 1,* 269–271.

FREDMAN, M. L. AND TARJAN, R. E. 1987. Fibonacci heaps and their uses in improved netweok optimization algorithms. *Journal of the ACM 34,* 596–615.

HENZINGER, M. R., KLEIN, PH., RAO, S., AND SUBRAMANIAN, S. 1997. Faster shortest-path algorithms for planar graphs. *Journal of Computer and System Sciences 55,* 1, part 1, 3–23.

KRAMER, M. R. AND VAN LEEUWEN, J. 1984. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. In *VLSI-Theory*, F. P. Preparata, Ed. Advances in Computing Research, vol. 2. JAI Press, Greenwich, Connecticut, 129–146.

RAGHAVAN, P. 1986. Randomized rounding and discrete ham-sandwich theorems: provably good algorithms for routing and packing problems. Ph.D. thesis, University of California, Berkeley, California. Report No. UCB/CSD 87/312.

REIF, J. H. 1983. Minimum $s - t$ cut of a planar undirected network in $O(n \log^2(n))$ time. *SIAM Journal on Computing 12,* 1, 71–81.

RIPPHAUSEN-LIPA, H., WAGNER, D., AND WEIHE, K. 1996. Linear-time algorithms for disjoint two-face paths problems in planar graphs. *International Journal of Foundations of Computer Science 7,* 2, 95–110.

ROBERTSON, N. AND SEYMOUR, P. D. 1995. Graph minors. XIII: the disjoint paths problem. *Journal of Combinatorial Theory, Series B 63,* 1, 65–110.

SCHRIJVER, A. 1994. Finding k disjoint paths in a directed planar graph. *SIAM Journal on Computing 23,* 4, 780–788.

SCHRIJVER, A. 2003. *Combinatorial optimization. Polyhedra and efficiency.* Algorithms and Combinatorics, vol. 24. Springer-Verlag, Berlin.

SUZUKI, H., AKAMA, T., AND NISHIZEKI, T. 1990. Finding Steiner forests in planar graphs. In *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).* 444–453.

TAKAHASHI, J.-Y., SUZUKI, H., AND NISHIZEKI, T. 1996. Shortest noncrossing paths in plane graphs. *Algorithmica 16,* 339–357.

VAN DER HOLST, H. AND DE PINA, J. C. 2002. Length-bounded disjoint paths in planar graphs. *Discrete Applied Mathematics 120,* 1–3, 251–261.