Four Shortest Vertex-Disjoint Paths in Planar Graphs*

Jeff Erickson and Yipu Wang

University of Illinois at Urbana-Champaign
{jeffe,ywang298}@illinois.edu

Abstract. Let G be an edge-weighted planar graph with $2k$ terminal vertices $s_1, t_1, \ldots, s_k, t_k$. The *minimum-sum vertex-disjoint paths* problem asks for a set of pairwise vertex-disjoint simple paths of minimum total length, where the ith path connects s_i to t_i. Even when all terminals lie on a single face, efficient algorithms for this problem are known only for fixed $k \le 3$. We describe the first polynomial-time algorithm for the case of four arbitrary terminal pairs on a single face.

1 Introduction

In the ***vertex-disjoint paths problem***, we are given a graph G along with k vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$, and we want to find k pairwise vertex-disjoint paths connecting each node s_i to the corresponding node t_i. The vertices $s_1, \ldots, s_k, t_1, \ldots, t_k$ are called ***terminals***. The vertex-disjoint paths problem is a special case of multi-commodity flows with applications in VLSI design [7, 19] and network routing [18, 22]. This problem is NP-hard [13] and remains so even if G is undirected planar [16] or if G is directed and $k = 2$ [6]. On the other hand, it can be solved in polynomial time if G is undirected and k is bounded [14, 20] or if G is directed acyclic and k is bounded [6]. Furthermore, the problem is fixed-parameter tractable with respect to the parameter k in directed planar graphs [5, 21].

We focus on an optimization version of the vertex-disjoint paths problem, where the goal is to minimize the total length of the paths. This version of the problem has also been considered in the context of network routing, where we want to minimize the amount of energy required to send packets [18, 22]. In the ***k-min-sum problem***, we are given a graph G, in which every edge e has a non-negative real length $\ell(e)$, and k pairs of vertices $(s_1, t_1), \ldots, (s_k, t_k)$. We want to find k vertex-disjoint paths $P_1, \ldots, P_k$ where each path P_i is a path from s_i to t_i and the total length $\sum_{i=1}^{k} \ell(P_i)$ is as small as possible. (Here $\ell(P_i) = \sum_{e \in P_i} \ell(e)$.)

Middendorf and Pfeiffer [16] proved that the k-min-sum problem is NP-hard when the parameter k is part of the input, even in undirected 3-regular plane graphs. Not much is known about the complexity of the planar k-min-sum problem for fixed k. In fact, no non-trivial algorithms or hardness results are known for either the 2-min-sum problem in directed planar graphs or the 5-min-sum problem in undirected planar graphs, even when all terminals are required to lie on a single face.

Polynomial-time algorithms for the planar k-min-sum problem are known for *arbitrary* k when all $2k$ terminals terminals lie on a single face, in one of two patterns.

* This work was partially supported by NSF grant CCF-1408763.

In a ***parallel*** instance, the terminals appear in cyclic order $s_1, \ldots, s_k, t_k, \ldots, t_1$, and an a ***serial*** instance, the terminals appear in cyclic order $s_1, t_1, s_2, t_2, \ldots, s_k, t_k$. Even in directed planar graphs, parallel instances of k-min-sum can be solved using a straightforward reduction to minimum-cost flows [9]; in fact, this special case can be solved in $O(kn)$ time. A recent algorithm of Borradaile, Nayyeri, and Zafarani [2] solves any serial instance of k-min-sum in an undirected planar graph in $O(kn^5)$ time.

If we allow arbitrary patterns of terminals, fast algorithms are known for only very small values of k. Kobayashi and Sommer [15] describe two algorithms, one running in $O(n^3 \log n)$ time when $k = 2$ and all four terminals are covered by at most two faces, the other running in $O(n^4 \log n)$ time when $k = 3$ when all terminals are incident to a single face. Colin de Verdière and Schrijver [4] describe an $O(kn \log n)$-time algorithm for directed planar graphs where all sources s_i lie on one face and all targets t_i lie on another face. Finally, if $k \leq 3$, every planar instance of k-min-sum with all terminals on the same face is either serial or parallel.

Zafarani [23] proved an important structural result for the planar k-min-sum problem. Consider an undirected edge-weighted plane graph G with vertices $s_1, t_1, \ldots, s_k, t_k$ on its outer face. Let $Q_1, Q_2, \ldots, Q_k$ be the shortest vertex-disjoint paths in G connecting all k terminal pairs, and let $P_1, P_2, \ldots, P_{k-1}$ be the shortest vertex-disjoint paths in G connecting every pair except s_k, t_k, where the subscript on each path indicates which terminals it connects. Zafarani's Structure Theorem states that if two paths P_i and Q_j cross, then $i = j$.

We describe the first polynomial-time algorithm to solve the 4-min-sum problem in undirected planar graphs with all eight terminals on a common face. If the given instance is parallel or serial, it can be solved using existing algorithms; otherwise, the terminals can be labeled $s_4, s_3, s_1, t_1, s_2, t_2, t_3, t_4$ in cyclic order around their common face. To solve these instances, our algorithm first computes a solution to the 3-min-sum problem for the terminal pairs s_1t_1, s_2t_2, s_4t_4, using the serial-instance algorithm of Borradaile *et al.* We identify a small set of key *anchor* vertices where the 3-min-sum solution intersects the 4-min-sum solution we want to compute. For each possible choice of anchor vertices, we connect these vertices to the terminals using parallel min-sum problems in three carefully constructed subgraphs of G. Overall, our algorithm runs in $O(n^6)$ time. Our characterization of the interaction between the 3-min-sum and 4-min-sum solutions, which both relies on and extends Zafarani's Structure Theorem [23], is the main technical contribution of the paper.

2 Preliminaries

For any integer N, let $\boldsymbol{[N]}$ denote the set $\{1, 2, \ldots, N\}$.

For any plane graph G, we write $\boldsymbol{\partial G}$ to denote the boundary of the outer face of G; we also informally call ∂G the boundary of G. Without loss of generality, we assume that ∂G is a simple cycle.

Our algorithms search for pairwise disjoint *walks* with minimum total length that connect corresponding terminals, rather than explicitly seeking simple paths. Because all edge lengths are non-negative, the shortest set of walks will of course consist of simple paths. The length of a walk w in an edge-weighted graph, which we denote $\boldsymbol{\ell(w)}$, is the sum of the lengths of its edges, with appropriate multiplicity of w is not a simple path. The total length of any set of walks $\mathcal{W}$, which we denote $\boldsymbol{\ell(\mathcal{W})} = \sum_{w \in \mathcal{W}} \ell(w)$, is just the sum of their lengths. Two walks *meet* or *touch* if they have at least one vertex in common.

For any path P and any vertices u and v on that path, we write $\boldsymbol{P[u,v]}$ to denote the subpath of P from u to v. Similarly, let $\boldsymbol{P[u,v)}$ denote the subpath of P from u to the predecessor of v, let $\boldsymbol{P(u,v]}$ denote the subpath of P from the successor of u to v, and let $\boldsymbol{P(u,v)}$ denote the subpath of P from the successor of u to the predecessor of v; these subpaths could be empty. The reversal of any path P is denoted $\boldsymbol{rev(P)}$. The concatenation of two paths P and P' is denoted $\boldsymbol{P \circ P'}$.

Our 4-min-sum algorithm relies on a black-box subroutine to solve parallel instances of 2-min-sum and 3-min-sum. As observed by van der Holst and de Pina, any parallel instance of k-min-sum can be solved in polynomial time by reduction to minimum-cost flow problem [9]. In fact, these instances can reduced in $O(n)$ time to a *planar* instance of minimum-cost flow, by replacing each vertex with a clockwise directed unit-capacity cycle, as described by Colin de Verdiére and Schrijver [4] and Kaplan and Nussbaum [12]. The resulting minimum-cost flow problem can then be solved $O(kn)$ time by performing k iterations of the classical successive shortest path algorithm [3,10,11], using the $O(n)$-time shortest-path algorithm of Henzinger *et al.* [8] at each iteration. We omit further details from this version of the paper.

To simplify our presentation, we assume that our given instance of 4-min-sum and every instance of 2-min-sum and 3-min-sum considered by our algorithm has a unique solution. If necessary, these uniqueness assumptions can be enforced with high probability using the isolation lemma of Mulmuley, Vazirani, and Vazirani [17]. We omit further details from this version of the paper.

3 Structure

Let G be an undirected plane graph with non-negative edge lengths, and let $s_4, s_3, s_1, t_1, s_2, t_2, t_3, t_4$ be eight distinct vertices in cyclic order around the outer face, as illustrated in Figure 1. Let $\mathcal{Q} = \{Q_1, \ldots, Q_4\}$ denote the unique optimal solution to this 4-min-sum instance, where each path Q_i connects s_i to t_i, and let $\mathcal{P} = \{P_1, P_2, P_4\}$ denote the unique optimal solution to the induced 3-min-sum problem that omits the demand pair s_3t_3, where again each path P_i connects s_i to t_i. We can compute $\mathcal{P}$ in $O(n^4 \log n)$ time using the algorithm of Kobayashi and Sommer [15], or in $O(n^5)$ time using the more general algorithm of Borradaile *et al.* [2].

The paths in $\mathcal{P}$ divide G into four regions, as shown in Figure 1(a). Let X be the unique region adjacent to all the paths in $\mathcal{P}$. For each index $i \neq 3$, let $\boldsymbol{C_i}$ denote the subpath of ∂G from s_i to t_i that shares no edges with X, let $\boldsymbol{R_i}$ denote the closed region bounded by P_i and C_i, and let $\boldsymbol{R_i^\circ}$ denote the half-open region $R_i \setminus P_i$.

3.1 How $\mathcal{P}$ intersects $\mathcal{Q}$

We begin by proving several simple structural properties of the 4-min-sum solution $\mathcal{Q}$ that will help us compute it quickly once we know the 3-min-sum solution $\mathcal{P}$. Figure 1(b) shows a typical structure for $\mathcal{Q}$.

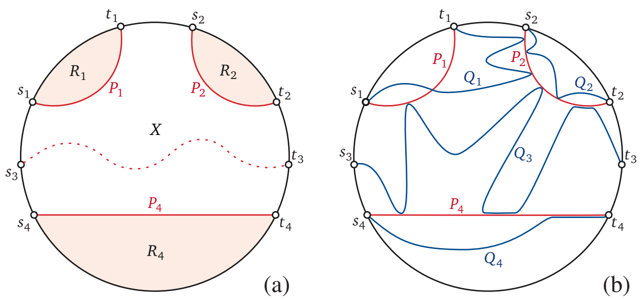

Fig. 1. (a) Terminals, paths in $\mathcal{P}$, and the regions they define. (b) Typical structure of $\mathcal{Q}$.

Lemma 3.1. *If P_i crosses Q_j, then $i = j$. In particular, Q_3 does not cross any path in $\mathcal{P}$.*

Proof. This is an immediate consequence Zafarani's structure theorem [23]. □

Lemma 3.2. *Q_1 and Q_2 do not meet P_4. Similarly, Q_4 does not meet P_1 or P_2.*

Proof. Q_3 separates s_1, t_1, s_2, t_2 from s_4 and t_4. Thus, Lemma 3.1 implies that Q_3 separates Q_1 and Q_2 from P_4, and Q_3 separates P_1 and P_2 from Q_4. □

Lemma 3.3. *Q_4 lies entirely in R_4.*

Proof. For the sake of argument, suppose Q_4 leaves R_4. Let q be any component of $Q_4 \setminus R_4^\circ$, as shown at the bottom of Figure 2. The endpoints x and y of q must lie on P_4; let p denote the subpath $P_4[x, y]$. Define two new paths $P_4' = P_4 \setminus p \cup q$ and $Q_4' = Q_4 \setminus q \cup p$. Both P_4' and Q_4' are walks from s_4 to t_4. Let $\mathcal{P}' = \{P_1, P_2, P_4'\}$ and $\mathcal{Q} = \{Q_1, Q_2, Q_3, Q_4'\}$.

Lemma 3.2 implies that $q \subseteq Q_4$ does not meet P_1 or P_2, so P_4' does not meet P_1 or P_2, which implies that the walks in $\mathcal{P}'$ are pairwise vertex-disjoint. On the other hand, subpath p lies inside the disk enclosed by $P_4' \cup C_4$, so Lemmas 3.1 and 3.2 imply that Q_4' does not meet Q_1, Q_2, or Q_3. It follows that the walks in $\mathcal{Q}'$ are also pairwise vertex-disjoint.

The unique optimality of $\mathcal{P}$ implies $\ell(\mathcal{P}) < \ell(\mathcal{P}')$, and the unique optimality of $\mathcal{Q}$ implies $\ell(\mathcal{Q}) < \ell(\mathcal{Q}')$. But $\ell(\mathcal{P}) + \ell(\mathcal{Q}) = \ell(\mathcal{P}') + \ell(\mathcal{Q}')$, so we have a contradiction. □

Lemma 3.4. *Every component of $Q_1 \setminus R_1^\circ$ meets P_2, and every component of $Q_2 \setminus R_2^\circ$ meets P_1.*

Proof. For the sake of argument, suppose some component q of of $Q_1 \setminus R_1^\circ$ does not meet P_2, as shown at the top of Figure 2. We derive a contradiction using a similar exchange argument to Lemma 3.3.

The endpoints x and y of q must lie on P_1; let p denote the subpath $P_1[x, y]$. Define two new paths $P_1' = P_1 \setminus p \cup q$ and $Q_1' = Q_1 \setminus q \cup p$. Clearly P_1' and Q_1' are both walks from s_1 to t_1. Let $\mathcal{P}' = \{P_1', P_2, P_4\}$ and $\mathcal{Q} = \{Q_1', Q_2, Q_3, Q_4\}$. Lemma 3.2 and our assumption that q does not meet P_2 imply that the walks in $\mathcal{P}'$ are pairwise vertex-disjoint. On the other hand, p lies in the disk enclosed by $P_1' \cup C_1$, which implies

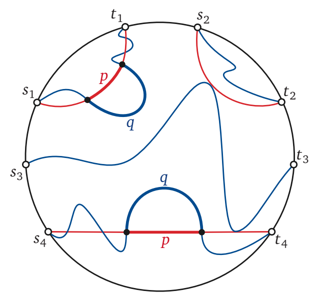

Fig. 2. An impossible configuration of optimal paths, for the proofs of Lemmas 3.3 and 3.4.

that the walks in $\mathcal{Q}'$ are also pairwise vertex-disjoint. The optimality of $\mathcal{P}$ implies that $\ell(\mathcal{P}) < \ell(\mathcal{P}')$, and the optimality of $\mathcal{Q}$ implies that $\ell(\mathcal{Q}) < \ell(\mathcal{Q}')$, but clearly $\ell(\mathcal{P})+\ell(\mathcal{Q}) = \ell(\mathcal{P}') + \ell(\mathcal{Q}')$, so we have a contradiction.

A symmetric argument proves that every component of $Q_2 \setminus R_2^\circ$ meets P_1. □

Lemma 3.5. *Either $Q_1 \subset R_1$ or $Q_2 \subset R_2$ or both.*

Proof. For the sake of argument, suppose Q_1 leaves R_1 and Q_2 leaves R_2. Let S_1 be the closed region bounded by $Q_1 \cup C_1$ and let S_2 be the closed region bounded by $Q_2 \cup C_2$. We call each component of $S_1 \setminus R_1^\circ$ a *left finger*, and each component of $S_2 \setminus R_2^\circ$ a *right finger*. Lemma 3.4 and the Jordan curve theorem imply that each finger is a topological disk that intersects both P_1 and P_2. Thus, the fingers can be linearly ordered by their intersections with P_1 from s_1 to t_1 (from bottom to top in Figure 3). Because Q_1 is a simple path, the fingers intersect Q_1 in the same order. Without loss of generality, suppose the last finger in this order is a right finger. Let s be the last left finger, and let s' be the right finger immediately after s.

Let w be the last node of P_1 (closest to t_1) that lies in s, and let y be the last node of P_2 (closest to t_2) that that lies in s'. We define four subpaths $p_1 = P_1[w, t_1]$, $q_1 = Q_1[w, t_1]$, $p_2 = P_2[s_2, y]$, and $q_2 = Q_2[s_2, y]$, as shown on the left of Figure 3. (Paths p_2 and q_2 could enclose more than one right finger.)

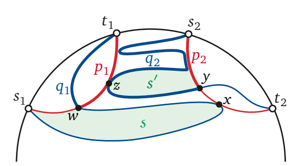

Fig. 3. Another impossible configuration of optimal paths, for the proof of Lemma 3.5.

Now exchange the subpaths $p_1 \leftrightarrow q_1$ and $p_2 \leftrightarrow q_2$ to define four new walks $P_1' = P_1 \setminus p_1 \cup q_1$, $Q_1' = Q_1 \setminus q_1 \cup p_1$, $P_2' = P_2 \setminus p_2 \cup q_2$, and $Q_2 = Q_2 \setminus q_2 \cup p_2$. Finally, let $\mathcal{P}' = \{P_1', P_2', P_4\}$ and $\mathcal{Q}' = \{Q_1', Q_2', Q_3, Q_4\}$. As in previous lemmas, we argue that $\mathcal{P}'$ and $\mathcal{Q}'$ are sets of vertex-disjoint walks.

Lemma 3.1 implies that Q_3 does not cross P_1 or P_2, and trivially Q_3 does not cross Q_1. Thus, none of the paths Q_3, P_4, Q_4 touches any of the paths p_1, q_1, p_2, q_2. It follows that P_4 does not touch either P'_1 or P'_2, and similarly, Q_3 and Q_4 does not touch either Q'_1 or Q'_2.

We define two more auxiliary nodes x and z, as shown on the right in Figure 3. Let x be the first vertex of P_2 also on Q_1. Vertex y must precede x on P_2, because $x \in s$ and $y \in s'$. Let z be the first vertex of P_1 also on q_2. Vertex w must precede z on P_1, because $w \in s$ and $z \in s'$.

Trivially, q_1 does not meet q_2, and $P_1 \setminus p_1$ does not meet $P_2 \setminus p_2$. Any left finger formed from q_1 must succeed s. Because s is the last left finger, q_1 does not form any left fingers and does not touch P_2. By definition, z is the first node of P_1 also on q_2. On the other hand, all vertices of $P_1 \setminus p_1$ (except w) precede w on P_1, which in turn strictly precedes z on P_1, so $P_1 \setminus p_1$ is disjoint from q_2. We conclude that P'_1 does not meet P'_2, implying that the walks in $\mathcal{P}'$ are vertex-disjoint:

Trivially, p_1 does not meet p_2, and $Q_1 \setminus q_1$ does not meet $Q_2 \setminus q_2$. Since q_1 does not meet P_2, x is the first vertex of P_2 also on $Q_1 \setminus q_1$. On the other hand, all vertices of p_2 (except y) precede y on P_2, which in turn strictly precedes x on P_2, so $Q_1 \setminus q_1$ is disjoint from p_2. Any right finger whose boundary contains a subpath of $Q_2 \setminus q_2$ must precede s', and any right finger that meets p_1 must succeed s. Because no right fingers lie strictly between s and s', the path $Q_2 \setminus q_2$ does not form any right fingers that meet p_1. We conclude that Q'_1 does not meet Q'_2, which implies that the walks in $\mathcal{Q}'$ are vertex-disjoint.

Finally, as usual, the optimality of $\mathcal{P}$ implies that $\ell(\mathcal{P}') < \ell(\mathcal{P}')$, and the optimality of $\mathcal{Q}$ implies that $\ell(\mathcal{Q}') < \ell(\mathcal{Q}')$, but clearly $\ell(\mathcal{P}) + \ell(\mathcal{Q}) = \ell(\mathcal{P}') + \ell(\mathcal{Q}')$. □

Without loss of generality, in light of Lemma 3.5, we assume for the rest of the paper that P_1 and Q_2 are disjoint, and thus $Q_2 \subset R_2$.

3.2 Subgraph Solutions

Our algorithm solves several parallel instances of k-min-sum inside certain subgraphs of G. To prove that our algorithm is correct, we need to argue that the subgraph solutions coincide exactly with portions of the desired global solution. As an intermediate step, we first show that the subgraph solutions interact with the global solution in a limited way.

Lemma 3.6. *Let $(G, \{s_i, t_i \mid i \in [k]\})$ be a planar instance of k-min-sum, with all terminals s_i and t_i on ∂G, whose unique solution is $\mathcal{Q} = \{Q_1, \ldots, Q_k\}$. Let S be a subset of $[k]$ such that the induced planar min-sum instance $(G, \{s_i, t_i \mid i \in S\})$ is parallel. Let H be a subgraph of G such that*

(1) $Q_i \cap H \neq \emptyset$ if and only if $i \in S$, and
(2) for all distinct $i, j \in S$, no component of $Q_i \cap H$ separates components of $Q_j \cap H$ from each other in H.

For each index $i \in S$, let u_i and v_i be vertices of $Q_i \cap \partial H$ such that $Q_i[u_i, v_i] \subseteq H$. Finally, suppose $(H, \{u_i, v_i \mid i \in S\})$ is a parallel planar min-sum instance, whose unique solution is $\Pi = \{\pi_i \mid i \in S\}$. Then for all indices $i, j \in S$, if $i \neq j$, then π_i does not cross Q_j.

Proof. First we establish some notation and terminology. Let $\kappa = |S|$, and re-index the terminals so that $S = [\kappa]$ and the counterclockwise order of terminals around the outer face of H is $u_1, \ldots, u_\kappa, v_\kappa, \ldots, v_1$. Fix an index i such that $1 \leq i < \kappa$, and consider the paths Q_i and π_{i+1}.

Let C ("ceiling") denote the path in ∂G from s_i to t_i that does not contain s_{i+1} or t_{i+1}, and let A be the closed region bounded by C and Q_i. A point in G is *above* Q_i if it lies in $A \setminus Q_i$ and *below* Q_i if it does not lie in A.

Similarly, let F ("floor") denote the path in ∂H from u_{i+1} to v_{i+1} that does not contain u_i or v_i, and let B be the closed region bounded by F and π_{i+1}. A point in H is *below* π_{i+1} if it lies in $B \setminus \pi_{i+1}$ and *above* π_{i+1} if it does not lie in B.

Paths Q_i and π_{i+1} also divide the interior of G into connected regions, exactly one of which has the entire path C on its boundary; call this region U. Finally, let Q'_i denote the unique path in G from s_i to t_i such that $C \cup Q'_i$ is the boundary of U. Every point on Q'_i lies on or above Q_i, and our assumption (2) implies that every point in $Q'_i \cap H$ lies on or above π_{i+1}. Thus, intuitively, Q'_i is the "upper envelope" of Q_i and π_{i+1}. In particular, $Q'_i = Q_i$ if and only if Q_i and π_{i+1} are disjoint.

Similarly, paths Q_i and π_{i+1} divide the interior of H into connected regions, exactly one of which contains F on its boundary; call this region L. Let π'_{i+1} denote the unique path in H from u_{i+1} to v_{i+1} such that $D \cup \pi'_{i+1}$ is the boundary of L. Assumption (2) implies that every point on π'_{i+1} lies on or below both π_{i+1} and Q_i. Thus, intuitively, π'_{i+1} is the "lower envelope" of Q_i and π_{i+1}. In particular, $\pi'_{i+1} = \pi_{i+1}$ if and only if Q_i and π_{i+1} are disjoint.

Each component of $Q'_i \setminus Q_i$ is an open subpath of π_{i+1} that lies entirely above Q_i and therefore is not contained in π'_{i+1}. Similarly, every component of $\pi'_{i+1} \setminus \pi_{i+1}$ is an open subpath of $Q_i \cap H$ that lies entirely below π_{i+1} and therefore is not contained in Q'_i. It follows that $\ell(Q'_i) + \ell(\pi'_{i+1}) \leq \ell(Q_i) + \ell(\pi_{i+1})$.

Finally, let $\mathcal{Q}' = \{Q'_1, \ldots, Q'_{\kappa-1}, Q_\kappa, \ldots, Q_k\}$ and $\Pi' = \{\pi_1, \pi'_2, \ldots, \pi'_\kappa\}$; see Figure 1 for an example of our construction.

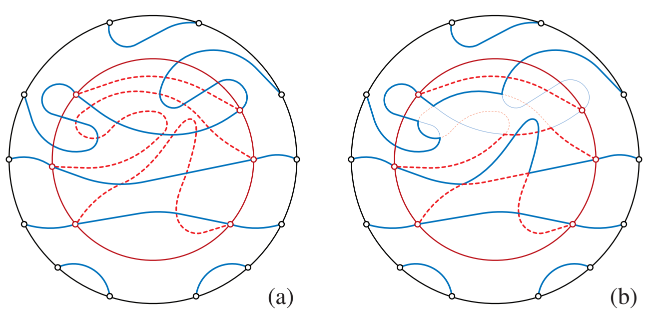

Fig. 4. Proof of Lemma 3.6. The inner red circle is ∂H. (a) The original disjoint paths $\mathcal{Q}$ (solid blue) and Π (dashed red). (b) The transformed disjoint paths $\mathcal{Q}'$ (solid blue) and Π' (dashed red).

Now suppose for the sake of argument that Q_i crosses π_{i+1} for some index i, or equivalently, that $\mathcal{Q}' \neq \mathcal{Q}$ and $\Pi' \neq \Pi$. As usual, to derive a contradiction, we need to show that $\mathcal{Q}'$ and Π' are sets of disjoint walks. The following case analysis implies that the walks in $\mathcal{Q}'$ are pairwise disjoint:

- None of the paths $Q_{\kappa+1}, \ldots, Q_k$ intersect H. On the other hand, for all $i < \kappa$, $Q'_i \setminus Q_i$ is a subset of π_{i+1} and therefore lies in H. Trivially, $Q_{\kappa+1}, \ldots, Q_k$ are disjoint from $Q_1, \ldots, Q_\kappa$. Thus, paths $Q'_1 \ldots, Q'_{\kappa-1}, Q_\kappa$ are disjoint from paths $Q_{\kappa+1}, \ldots, Q_k$.
- Q_κ lies entirely below $Q_{\kappa-1}$ and therefore entirely below $Q'_{\kappa-1}$.
- Now consider any point $x \in Q'_i$, for any index $1 \le i < \kappa - 1$. Point x lies on or above Q_i (because every point in Q'_i lies on or above Q_i), and therefore lies above Q_{i+1}. So we must have $x \in \pi_{i+2}$ and therefore $x \in H$. But because $x \in Q'_i \cap H$, x lies either on or above π_{i+1}, and therefore lies above π_{i+2}. So x cannot lie on Q'_{i+1}. We conclude that Q'_i and Q'_{i+1} are disjoint.

Similar case analysis implies that the walks in Π' are pairwise disjoint:

- π_1 lies entirely above π_2 and therefore entirely above π'_2.
- Now consider any point $x \in \pi'_{i+1}$, for any index $1 < i < \kappa$. Point x lies on or below Q_i, and therefore below Q_{i-1}. On the other hand, x lies on or below π_{i+1}, and therefore lies below π_i. So x cannot lie in π'_i. We conclude that π'_i and π'_{i+1} are disjoint.

The unique optimality of Π and $\mathcal{Q}$ implies $\ell(\Pi) < \ell(\Pi')$ and $\ell(\mathcal{Q}) < \ell(\mathcal{Q}')$. On the other hand, we immediately have

$$\begin{aligned} \ell(\Pi) + \ell(\mathcal{Q}) &= \ell(\pi_1) + \sum_{i=1}^{\kappa-1} \big(\ell(Q_i) + \ell(\pi_{i+1})\big) + \sum_{i=\kappa}^{k} \ell(Q_i) \\ &\le \ell(\pi_1) + \sum_{i=1}^{\kappa-1} \big(\ell(Q'_i) + \ell(\pi'_{i+1})\big) + \sum_{i=\kappa}^{k} \ell(Q_i) \\ &= \ell(\Pi') + \ell(\mathcal{Q}'), \end{aligned}$$

giving us a contradiction.

We conclude that π_i does not cross Q_{i-1} for any index i. It follows immediately that π_i does not cross (in fact, does not *touch*) any Q_j such that $j < i - 1$. A symmetric argument implies that π_i does not cross any Q_j such that $j > i$. □

4 Algorithm

Now we are finally ready to describe our algorithm for computing $\mathcal{Q}$ given $\mathcal{P}$. We define five ***anchor vertices*** as follows; see Figure 5.

- If Q_1 meets P_2, then a is the first vertex of Q_1 that is also on P_2, and b is the first vertex in the suffix $P_2(a, t_2]$ that is also on Q_2; otherwise, $a = t_1$ and $b = s_2$.
- If Q_3 meets P_2, then c is the first vertex in their intersection; otherwise, $c = t_3$.
- If P_4 meets the prefix $Q_3[s_3, c)$, then d is the last vertex in their intersection; otherwise, $d = s_4$.
- Finally, e is the first vertex of the suffix $P_4(d, t_4]$ that is also on Q_4.

We also split each path Q_i into a prefix Q_i^s and a suffix Q_i^t that meet at a single vertex. Specifically, we split Q_1 at a, we split Q_2 at b, we split Q_3 at c, and we split Q_4 at e. Thus, for example, $Q_1^s = Q_1[s_1, a]$ and $Q_1^t = Q_1[a, t_1]$.

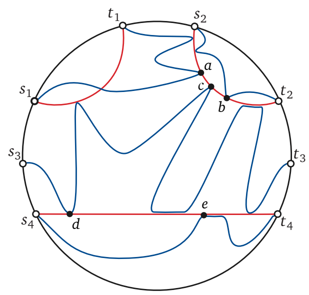

Fig. 5. Anchor vertices a, b, c, d, e.

Now suppose that we know the locations of the anchor vertices a, b, c, d, and e. Our algorithm computes $\mathcal{Q}$ in three phases; each phase solves a parallel instance of the k-min-sum problem (with $k = 2$ or $k = 3$) in a subgraph of G in $O(n)$ time, via minimum-cost flows. The subpaths of $\mathcal{Q}$ computed in each phase are shown in Figure 6.

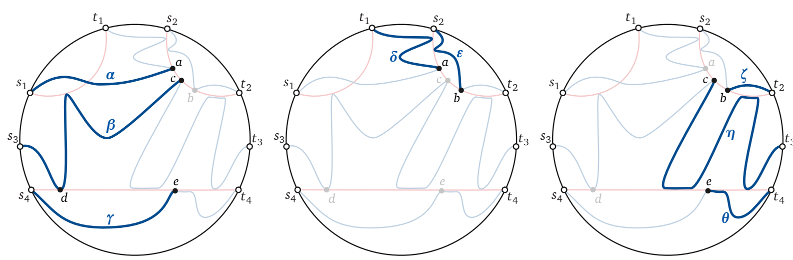

Fig. 6. Subpaths of $\mathcal{Q}$ computed by the three phases of our algorithm.

4.1 Phase 1: α, β, and γ

Let H_1 be the subgraph of G obtained by deleting every vertex in R_2 except a and c, every edge incident to s_4 or e outside of R_4, and every vertex of $P_4(d, t_4]$ except e. In the first phase, we compute the shortest set of vertex-disjoint paths in H_1 from s_1 to a, from s_3 to c, and from s_4 to e. Call these paths α, β, and γ, respectively.

Lemma 4.1. *$\alpha = Q_1^s$, $\beta = Q_3^s$, and $\gamma = Q_4^s$.*

Proof. For the sake of argument, suppose $(\alpha, \beta, \gamma) \neq (Q_1^s, Q_3^s, Q_4^s)$, and define a new set of walks $\mathcal{Q}' := \{\alpha \circ Q_1^t,\ Q_2,\ \beta \circ Q_3^t,\ \gamma \circ Q_4^t\}$. The following exhaustive case analysis implies that the walks in $\mathcal{Q}'$ are vertex-disjoint.

- Paths α, β, and γ are disjoint by definition.
- Similarly, Q_1^t, Q_2, Q_3^t, Q_4^t are subpaths of paths in $\mathcal{Q}$ and thus are disjoint by definition.
- P_2 separates Q_2 from α, β, and γ.
- Lemma 3.6 implies that β and γ do not cross Q_1^s, and therefore do not touch Q_1^t.
- Lemma 3.6 also implies that α does not cross Q_3^s, and therefore does not touch Q_3^t.
- Lemma 3.6 also implies that α and β do not cross Q_4^s, and therefore do not touch Q_4^t.
- Finally, if $d = s_4$, then the definition of H_1 implies that γ does not leave R_4° except at s_4 and e, so Lemma 3.1 implies that γ is disjoint from Q_3^t. On the other hand, if $d \neq s_4$, then Lemma 3.6 implies that γ does not cross $Q_3[s_3, d]$; on the other hand, Q_3^t does not meet $Q_3[s_3, d]$. The definition of H_1 implies that γ does not cross the path $P_4[d, t_4]$ and only meets it at d or e; on the other hand, neither d not e are on Q_3^t. Since $Q_3[s_3, d] \circ P_4[d, t_4]$ separates γ from Q_3^t, we conclude that Q_3^t and γ are disjoint.

Because the walks in $\mathcal{Q}'$ are vertex-disjoint, the unique optimality of $\mathcal{Q}$ implies that $\ell(\mathcal{Q}) < \ell(\mathcal{Q}')$.

On the other hand, the lemmas in Section 3.1 and the definitions of the anchor vertices imply that Q_1^s, Q_3^s, and Q_4^s are indeed paths in H_1 between the appropriate terminals. Moreover, Q_1^s, Q_3^s, and Q_4^s are vertex-disjoint, because they are subpaths of the disjoint paths in $\mathcal{Q}$. Thus, the unique optimality of $\{\alpha, \beta, \gamma\}$ implies that $\ell(\alpha) + \ell(\beta) + \ell(\gamma) < \ell(Q_1^s) + \ell(Q_3^s) + \ell(Q_4^s)$. It follows that $\ell(\mathcal{Q}') < \ell(\mathcal{Q})$, giving us the desired contradiction. □

4.2 Phase 2: δ and ε

If Q_1 and P_2 are disjoint, let $\delta = t_1$ and $\varepsilon = s_2$. Otherwise, let H_2 be the subgraph of G obtained by deleting every vertex of $P_2(a, t_2]$ except b, all edges incident to b that leave R_2, and every vertex of α except a. In the second phase, our algorithm computes the shortest vertex-disjoint paths in H_2 from t_1 to a and from s_2 to b. Call these paths δ and ε, respectively.

Lemma 4.2. *$rev(\delta) = Q_1^t$ and $\varepsilon = Q_2^s$.*

Proof. The lemma is obvious if Q_1 and P_2 are disjoint, so assume otherwise.

For the sake of argument, suppose $(rev(\delta), \varepsilon) \neq (Q_1^t, Q_2^s)$, and let $\mathcal{Q}' = \{Q_1^s \circ rev(\delta), \varepsilon \circ Q_2^t, Q_3, Q_4\}$. The following exhaustive case analysis implies that the walks in $\mathcal{Q}'$ are pairwise disjoint.

- δ and ε are disjoint by definition.
- Q_1^s, Q_2^t, Q_3, and Q_4 are disjoint by definition of $\mathcal{Q}$.
- Lemma 3.6 implies that δ does not cross Q_2^s, and therefore does not touch Q_2^t.
- The path $\alpha \circ P_2[a, t_2]$ separates δ and ε from Q_3 and therefore from Q_4.
- Lemma 4.1 implies that $Q_1^s \cap V(H_2) = \{a\}$. It follows that ε does not touch Q_1^s.

The unique optimality of $\mathcal{Q}$ now implies that $\ell(\mathcal{Q}) < \ell(\mathcal{Q}')$.

On the other hand, the lemmas in Section 3.1 and the definitions of the anchor vertices imply that Q_1^t and Q_2^s are vertex-disjoint paths in H_2 between the appropriate terminals. Thus, the unique optimality of $\{\delta, \varepsilon\}$ implies that $\ell(Q_1^t) + \ell(Q_2^s) > \ell(\delta) + \ell(\varepsilon)$, and therefore $\ell(\mathcal{Q}) > \ell(\mathcal{Q}')$, giving us the desired contradiction. □

4.3 Phase 3: ζ, η, and θ

Finally, let H_3 be the subgraph of G obtained by deleting all vertices in $\alpha \cdot rev(\delta)$, all vertices in $\beta[s_3, b)$, all vertices in $\gamma[s_4, e)$, and all vertices in $\varepsilon[s_2, b)$. The last phase of our algorithm computes the shortest vertex-disjoint paths in H_3 from b to t_2, from c to t_3, and from e to t_4. Call these paths ζ, η, and θ, respectively.

Lemma 4.3. *$\zeta = Q_2^t$, $\eta = Q_3^t$, and $\theta = Q_4^t$.*

Proof. Suppose, for the sake of argument, that $(\zeta, \eta, \theta) \neq (Q_2^t, Q_3^t, Q_4^t)$, and let $\mathcal{Q}' := \{Q_1,\ Q_2^s \circ \zeta,\ Q_3^s \circ \eta,\ Q_4^s \circ \theta\}$. As usual, exhaustive case analysis implies that the walks in $\mathcal{Q}'$ are pairwise disjoint. Several cases rely on Lemmas 4.1 and 4.2, which imply that $\alpha \circ rev(\delta) = Q_1, \beta = Q_3^s, \gamma = Q_4^s$, and $\varepsilon = Q_2^s$.

- ζ, η, and θ are disjoint by definition.
- Q_1, Q_2^s, Q_2^s, and Q_2^s are disjoint by definition of $\mathcal{Q}$.
- Q_1 is disjoint from H_3 and thus disjoint from ζ, η, and θ.
- $Q_2^s \cap H_3 = \{b\}$, so Q_2^s is disjoint from η and θ.
- $Q_3^s \cap H_3 = \{c\}$, so Q_3^s is disjoint from ζ and θ.
- $Q_4^s \cap H_3 = \{e\}$, so Q_3^s is disjoint from ζ and η.

The unique optimality of $\mathcal{Q}$ now implies that $\ell(\mathcal{Q}) < \ell(\mathcal{Q}')$.

On the other hand, Q_2^t, Q_3^t, and Q_4^t are paths between appropriate terminals in H_3. Thus, the unique optimality of $\{\zeta, \eta, \theta\}$ implies that $\ell(Q_2^t) + \ell(Q_3^t) + \ell(Q_3^t) > \ell(\zeta) + \ell(\eta) + \ell(\theta)$, and therefore $\ell(\mathcal{Q}) > \ell(\mathcal{Q})$, giving us the desired contradiction. □

4.4 Summary

Finally, we summarize our overall 4-min-sum algorithm. First, in a preprocessing phase, we compute $\mathcal{P}$ using the algorithm of Kobayashi and Sommer [15]. Then for all possible choices for the anchor vertices a, b, c, d, e, we compute the paths $\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \eta, \theta$ as described in the previous sections, first under the assumption that $Q_2 \subset R_2$, and then under the symmetric assumption that $Q_1 \subset R_1$ (mirroring the definitions of the anchor vertices and the paths). Altogether, we compute the solutions to $O(n^5)$ parallel instances of 2-min-sum and 3-min-sum, each in $O(n)$ time via minimum-cost flows. Thus, the overall running time of our algorithm is $O(n^6)$.

References

1. Ravindra K. Ahuja, Thomas L. Magnanti, and James Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
2. Glencora Borradaile, Amir Nayyeri, and Farzad Zafarani. Towards single face shortest vertex-disjoint paths in undirected planar graphs. *Proc. 23rd Ann. Europ. Symp. Algorithms*, 227–238, 2015. Lecture Notes Comput. Sci. 9294, Springer.
3. Robert G. Busacker and Paul J. Gowen. A procedure for determining minimal-cost network flow patterns. Technical Paper ORO-TP-15, Oper. Res. Office, Johns Hopkins U., November 1960. ⟨http://www.dtic.mil/docs/citations/AD0249662⟩.

4. Éric Colin de Verdière and Alexander Schrijver. Shortest vertex-disjoint two-face paths in planar graphs. *ACM Trans. Algorithms* 7(2):19:1–19:12, 2011.
5. Marek Cygan, Dániel Marx, Marcin Pilipczuk, and Michał Pilipczuk. The planar directed k-vertex-disjoint paths problem is fixed-parameter tractable. *Proc. IEEE 54th Ann. Symp. Found. Comput. Sci.*, 197–206, 2013.
6. Steven Fortune, John Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theoret. Comput. Sci.* 10((2)):111–121, 1980.
7. András Frank. Packing paths, cuts and circuits - a survey. *Paths, Flows, and VLSI-Layout*, 49–100, 1990. Springer.
8. Monika R. Henzinger, Philip Klein, Satish Rao, and Sairam Subramanian. Faster shortest-path algorithms for planar graphs. *J. Comput. Syst. Sci.* 55:3–23, 1997.
9. Hein van der Holst and José Coelho de Pina. Length-bounded disjoint paths in planar graphs. *Discrete Appl. Math.* 120(1–3):251–261, 2002.
10. Masao Iri. A new method of solving transportation-network problems. *J. Oper. Res. Soc. Japan* 3:27–87, 1960. ⟨http://www.orsj.or.jp/~archive/menu/01_03.html⟩.
11. William S. Jewell. Optimal flow through networks. Interim Tech. Rep. 8, Oper. Rec. Center, MIT, 1958. Cited in [1].
12. Haim Kaplan and Yahav Nussbaum. Maximum flow in directed planar graphs with vertex capacities. *Algorithmica* 61(1):174–189, 2011.
13. Richard Karp. On the computational complexity of combinatorial problems. *Networks* 5:45–68, 1974.
14. Ken-ichi Kawarabayashi and Paul Wollan. A shorter proof of the graph minor algorithm: The unique linkage theorem. *Proc. 42nd Ann. ACM Symp. Theory Comput.*, 687–694, 2010.
15. Yusuke Kobayashi and Christian Sommer. On shortest disjoint paths in planar graphs. *Discrete Optimization* 7(4):234–245, 2010.
16. Matthias Middendorf and Frank Pfeiffer. On the complexity of the disjoint paths problem. *Combinatorica* 13(1):97–107, 1993.
17. Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica* 7(1):105–113, 1987.
18. Richard G. Ogier, Vladislav Rutenburg, and Nachum Shacham. Distributed algorithms for computing shortest pairs of disjoint paths. *IEEE Trans. Inf. Theory* 39((2)):443–455, 1993.
19. Neil Robertson and Paul D. Seymour. An outline of a disjoint paths algorithm. *Paths, Flows, and VLSI-Layout*, 267–292, 1990. Springer.
20. Neil Robertson and Paul D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Combin. Theory, Ser. B* 63(1):65–110, 1995.
21. Alexander Schrijver. Finding k disjoint paths in a directed planar graph. *SIAM J. Comput.* 23((4)):780–788, 1994.
22. Anand Srinivas and Eytan Modiano. Finding minimum energy disjoint paths in wireless ad-hoc networks. *Wireless Networks* 11(4):401–417, 2005.
23. Farzad Zafarani. *A Structural Theorem For Shortest Vertex-Disjoint Paths Computation in Planar Graphs*. Master's thesis, Oregon State University, May 2016. ⟨https://ir.library.oregonstate.edu/xmlui/handle/1957/59656⟩.