Connectivity of Graphs Under Edge Flips

Norbert Zeh

Faculty of Computer Science, Dalhousie University,
6050 University Ave, Halifax, NS B3H 2Y5, Canada

nzeh@cs.dal.ca

Abstract

We study the following problem: Given a set V of n vertices and a set $\mathcal{E}$ of m edge pairs, we define a graph family $\mathcal{G}(V, \mathcal{E})$ as the set of graphs that have vertex set V and contain exactly one edge from every pair in $\mathcal{E}$. We want to find a graph in $\mathcal{G}(V, \mathcal{E})$ that has the minimal number of connected components. We show that, if the edge pairs in $\mathcal{E}$ are non-disjoint, the problem is NP-hard. This is true even if an edge is not allowed to appear in more than two edge pairs and the union of the graphs in $\mathcal{G}(V, \mathcal{E})$ is planar. if the edge pairs are disjoint, we provide an $\mathcal{O}(n^2 m)$-time algorithm that finds a graph in $\mathcal{G}(V, \mathcal{E})$ with the minimal number of connected components. Our proof of the latter statement is obtained by flipping edges in the graphs in $\mathcal{G}(V, \mathcal{E})$, where the flip of an edge e in a graph $G \in \mathcal{G}(V, \mathcal{E})$ removes e from G and inserts the other edge in the pair in $\mathcal{E}$ that contains e. We explore also the question whether any graph in $\mathcal{G}(V, \mathcal{E})$ can be transformed into any other graph in $\mathcal{G}(V, \mathcal{E})$ using edge flips while guaranteeing an upper bound on the number of connected components of every intermediate graph.

1 Introduction

Description of the problem. Given a set V of n vertices and a set $\mathcal{E}$ of m edge pairs, we define a graph family $\mathcal{G}(V, \mathcal{E})$ as the set of graphs that have vertex set V and contain exactly one edge from every pair in $\mathcal{E}$. The *maximal connectivity problem* (MCP) is the problem of finding a graph G^* in $\mathcal{G}(V, \mathcal{E})$ that has the minimal number of connected components. We call such a graph G^* *maximally connected* or *maximal*. Edelsbrunner [5] proposed MCP as a graph-theoretic formulation of a problem arising in the repair of self-intersections of triangulated surfaces [6]. We show that, if the edge pairs in $\mathcal{E}$ are non-disjoint, MCP is NP-hard. If the edge pairs are disjoint, we provide a polynomial-time solution for MCP. We obtain a maximal graph in $\mathcal{G}(V, \mathcal{E})$ by starting with an arbitrary graph in $\mathcal{G}(V, \mathcal{E})$ and making only local changes, so-called edge flips, that do not increase the number of connected components in the graph. We study also the question whether any graph in $\mathcal{G}(V, \mathcal{E})$ can be transformed into any other graph in $\mathcal{G}(V, \mathcal{E})$ using edge flips while guaranteeing an upper bound on the number of connected components in every intermediate graph.

Motivation and related work. Edge flips have received considerable attention, particularly in the context of geometric graphs such as triangulations and pseudo-triangulations of planar point sets [1, 3, 7, 8, 9, 11, 12]. The reason is that they are combinatorially interesting and potentially lead to efficient algorithms for solving certain optimization problems on graphs. In general, one considers a family $\mathcal{G}$ of graphs that have the same vertex set and usually the same number of edges.

An edge flip removes an edge e from a graph G in $\mathcal{G}$ and replaces it with another edge e' so that the resulting graph is also in $\mathcal{G}$. (Other types of flips that introduce or remove edges have been studied, for instance, in [1, 3].) Often, the structure of the graphs in $\mathcal{G}$ guarantees that every flip happens in a small subgraph of G; that is, flips are local transformations. For example, the flip of an edge e in a triangulation of a planar point set replaces edge e with the other diagonal of the quadrilateral that is the union of the two triangles on either side of e. If the structure of the graph does not guarantee the locality of flips, we may explicitly restrict our attention to local flips. If we have a certain quality measure of the graphs in $\mathcal{G}$ such as Delaunayhood (for triangulations) or the number of faces (for pseudo-triangulations), it is interesting to ask whether a globally optimal graph in $\mathcal{G}$ can be obtained by making only local changes that improve the quality of the graph. If the answer is affirmative and the number of required flips is small, efficient algorithms result because local transformations of the graphs can often be implemented efficiently.

A rich literature deals with edge flips in geometric and planar graphs [1, 2, 3, 7, 8, 9, 11, 12, 13, 14, 15, 16]. We only discuss a few of these results here. A by now classical result is that $\Theta(n^2)$ Delaunay flips are sufficient and necessary in the worst case to transform any triangulation of a point set P into the Delaunay triangulation of P [8], where a Delaunay flip replaces an edge that violates the empty-circle property of the Delaunay triangulation [4]. In [12], it is shown that the same bound holds for transforming any two triangulations into each other using arbitrary diagonal flips. More precisely, they show that $\Theta(n+k^2)$ flips are required to transform any two triangulations of a simple polygon with k reflex vertices into each other, and $\Theta(kn)$ flips are required to transform any two triangulations of a planar point set with k convex layers into each other. In [9], simultaneous flipping of multiple edges is allowed and the flip distance between any two triangulations using such parallel flips is shown to be $\Theta(n)$. Aichholzer et al. [1, 3] prove that, by allowing so-called edge-removing flips, the number of flips required to transform any minimum pseudo-triangulation into any other minimum pseudo-triangulation of a point set can be reduced from $\Theta(n^2)$ to $\mathcal{O}(n \log^2 n)$; any pseudo-triangulation can be made minimum using $\mathcal{O}(n)$ of these flips. Negami [15] studies diagonal flips in triangulated planar graphs; that is, only the topology, but not the geometry of the graph matters in this case. Aichholzer et al. [2] study local transformations of non-crossing spanning trees of planar point sets, including a continuous version of an edge flip, called an edge-slide, and prove upper bounds on the number of such transformations required to obtain a minimum spanning tree from any non-crossing spanning tree. Other relevant papers include [7, 11, 13, 14, 16], which study the expected length of flip sequences in the randomized incremental construction of Delaunay triangulations in two and higher dimensions.

Terminology and notation. We denote the number of connected components of a graph G by $\omega(G)$. We define $\tilde{\omega}(\mathcal{G}(V,\mathcal{E})) = \min\{\omega(G) : G \in \mathcal{G}(V,\mathcal{E})\}$; in particular, a graph $G^* \in \mathcal{G}(V,\mathcal{E})$ is maximal if $\omega(G^*) = \tilde{\omega}(\mathcal{G}(V,\mathcal{E}))$. We say that a family $\mathcal{G}(V,\mathcal{E})$ is *k-thick* if every edge appears in at most k pairs in $\mathcal{E}$; in particular, $\mathcal{G}(V,\mathcal{E})$ is 1-thick if the edge pairs in $\mathcal{E}$ are pairwise disjoint. We define *k-MCP* to be MCP restricted to k-thick families; *planar MCP* is MCP restricted to families $\mathcal{G}(V,\mathcal{E})$ such that the graph $(V, \bigcup_{P\in\mathcal{E}} P)$ is planar.

For a 1-thick family $\mathcal{G}(V,\mathcal{E})$, the *flip* of an edge e in a graph $G \in \mathcal{G}(V,\mathcal{E})$ removes edge e from G and replaces it with the other edge $\bar{e}$ in the edge pair $P \in \mathcal{E}$ that contains e. We call $\bar{e}$ the *complementary edge* or *complement* of e and denote the graph $(V, (E(G) \setminus e) \cup \{\bar{e}\})$ obtained by flipping edge e in G as $G\langle e\rangle$. More generally, we denote the graph obtained from G by flipping edges $e_1, \ldots, e_q$ as $G\langle e_1, \ldots, e_q\rangle$. We call the flip of an edge e *splitting*, *stable*, or *merging* depending on

whether $\omega(G\langle e\rangle)$ is greater than, equal to, or less than $\omega(G)$. A flip sequence $e_1, \ldots, e_q$ is merging if every flip in the sequence is stable or merging and $\omega(G\langle e_1, \ldots, e_q\rangle) < \omega(G)$. A merging flip sequence $e_1, \ldots, e_q$ is *maximizing* if $G\langle e_1, \ldots, e_q\rangle$ is maximal.

In Section 5, we consider $\mathcal{G}(V, \mathcal{E})$ to be itself a graph whose vertices are the graphs in $\mathcal{G}(V, \mathcal{E})$ and such that there is an edge between two graphs G_1 and G_2 if $G_2 = G_1\langle e\rangle$, for some edge $e \in G_1$. We use $\mathcal{G}(V, \mathcal{E}, k)$ to denote the subgraph of $\mathcal{G}(V, \mathcal{E})$ induced by all vertices $G \in \mathcal{G}(V, \mathcal{E})$ such that the graph G has at most k connected components.

Our results. We prove the following results:

- Planar k-MCP is NP-hard, for any $k > 1$. (Section 2)
- Every graph in a 1-thick family $\mathcal{G}(V, \mathcal{E})$ has a maximizing sequence of length at most $n - 1$. (Section 3)
- For any non-maximal graph G in a 1-thick family $\mathcal{G}(V, \mathcal{E})$, a merging sequence of at most $n-1$ flips can be found in $\mathcal{O}(nm)$ time. This implies that we can find a maximizing sequence of at most $n - 1$ flips in $\mathcal{O}(n^2 m)$ time. (Section 4)
- For any 1-thick family $\mathcal{G}(V, \mathcal{E})$ and any $k > \tilde{\omega}(\mathcal{G}(V, \mathcal{E}))$, the graph $\mathcal{G}(V, \mathcal{E}, k)$ is connected and has diameter at most $m + n - 1$; that is, for any two graphs $G_1, G_2 \in \mathcal{G}(V, \mathcal{E}, k)$, there exists a sequence $e_1, e_2, \ldots, e_q$ of at most $m + n - 1$ edges such that $G_1\langle e_1, e_2, \ldots, e_q\rangle = G_2$ and $\omega(G_1\langle e_1, e_2, \ldots, e_i\rangle) \leq k$, for all $1 \leq i \leq q$. $\mathcal{G}(V, \mathcal{E}, \tilde{\omega}(\mathcal{G}(V, \mathcal{E})))$ is not necessarily connected. (Section 5)

2 NP-Hardness of Planar k-MCP

Our proof that planar k-MCP is NP-hard for $k > 1$ uses a linear-time reduction from 3-SAT to planar 2-MCP. First we recall the necessary terminology. Given a Boolean variable x, we denote its negation as $\bar{x}$. A *literal* is a Boolean variable or its negation. A *clause* is the disjunction of literals: $C = \lambda_1 \vee \lambda_2 \vee \cdots \vee \lambda_k$. A Boolean formula F is in *conjunctive normal form* (CNF) if it is of the form $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where $C_1, \ldots, C_m$ are clauses. Formula F is in 3-CNF if every clause C_i, $1 \leq i \leq m$, contains exactly three literals. In this case, we denote the literals in C_i as $\lambda_{i,1}$, $\lambda_{i,2}$, and $\lambda_{i,3}$. We denote the Boolean variables in F as $x_1, \ldots, x_n$. It is well-known that the problem of deciding whether a given formula in 3-CNF is satisfiable, 3-*SAT*, is NP-complete [10]. Hence, if we can provide a polynomial-time reduction from 3-SAT to MCP, MCP is NP-hard.

An important element used in a number of constructions in this paper is the "connector graph" shown in Figure 1. This graph is planar. Its edges are grouped into *disjoint* edge pairs as indicated

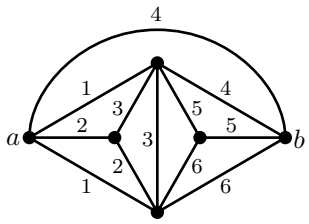

Figure 1: The connector graph. Two edges with the same number belong to the same edge pair.

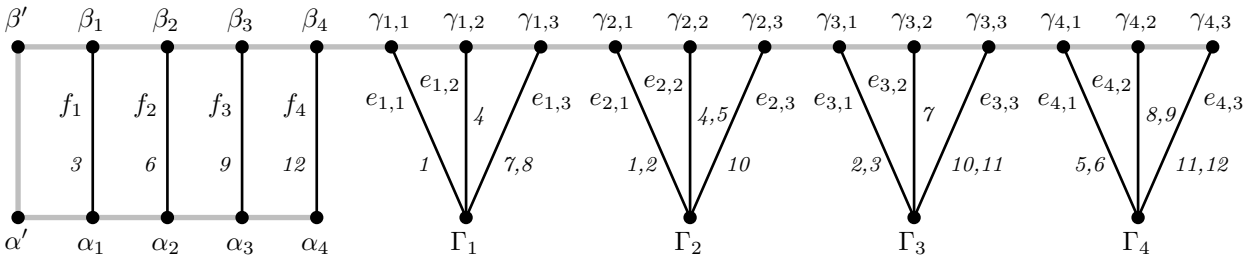

Figure 2: The graph $G'(F)$ for the formula $F = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \wedge (x_1 \vee x_3 \vee \bar{x}_4) \wedge (x_2 \vee x_3 \vee x_4)$. Regular edges are labelled with their names. The small italic labels identify the edge pairs that contain each edge.

by the numbering in Figure 1. It is easy to verify that, no matter which edge we choose from each edge pair, the resulting subgraph is connected. Hence, we can distinguish two of the vertices, a and b, and think of the graph as a "permanent" edge between vertices a and b; that is, this edge has to be present in every graph in the graph family. We will represent such permanent edges as fat grey edges in subsequent figures.

Given a formula F in 3-CNF with n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$, we construct a graph $G'(F)$ and assign its edges to appropriate edge pairs to obtain a family $\mathcal{G}(F)$ of subgraphs of $G'(F)$. We ensure that no edge is in more than two pairs, that $G'(F)$ is planar, and that $\mathcal{G}(F)$ contains a connected graph if and only if F is satisfiable. For every literal λ, let $\kappa(\lambda)$ be the number of clauses containing λ. For a variable x_i, we define $\kappa^*(x_i) = |\kappa(x_i) - \kappa(\bar{x}_i)|$. Let $\kappa^* = \sum_{i=1}^{n} \kappa^*(x_i)$. The vertex set of $G'(F)$ contains vertices $\Gamma_1, \ldots, \Gamma_m$, one per clause; vertices $\gamma_{i,j}$, $1 \le i \le m$ and $1 \le j \le 3$, one per literal $\lambda_{i,j}$; and vertices $\alpha_1, \ldots, \alpha_{k^*}, \beta_1, \ldots, \beta_{k^*}, \alpha', \beta'$. The vertices of $G'(F)$, excluding vertices $\Gamma_1, \ldots, \Gamma_m$, are connected using permanent edges to form a chain as shown in Figure 2. Besides these permanent edges, graph $G'(F)$ contains regular edges $e_{i,j} = (\Gamma_i, \gamma_{i,j})$, $1 \le i \le m$ and $1 \le j \le 3$, and $f_k = (\alpha_k, \beta_k)$, $1 \le k \le \kappa^*$. We refer to an edge $e_{i,j}$ as a *literal edge* and to an edge f_k as a *dummy edge*. Graph $G'(F)$ is obviously planar.

Next we group literal and dummy edges into pairs so that, in every graph in $\mathcal{G}(F)$, a literal edge $e_{i,j}$ is present if and only if every edge $e_{i',j'}$ with $\lambda_{i,j} = \lambda_{i',j'}$ is present and every edge $e_{i'',j''}$ with $\lambda_{i,j} = \bar{\lambda}_{i'',j''}$ is absent. Hence, the presence and absence of edges in a graph in $\mathcal{G}(F)$ corresponds to a truth assignment to the variables $x_1, \ldots, x_n$. Consider a variable x_k and the literals $\lambda_{i_1,j_1}, \ldots, \lambda_{i_q,j_q}$ such that $\lambda_{i_h,j_h} = x_k$ or $\lambda_{i_h,j_h} = \bar{x}_k$, for all $1 \le h \le q$. We assume w.l.o.g. that $\lambda_{i_1,j_1} = \cdots = \lambda_{i_r,j_r} = x_k$ and $\lambda_{i_{r+1},j_{r+1}} = \cdots = \lambda_{i_q,j_q} = \bar{x}_k$, for some $1 \le r \le q$. We also assume that $\kappa^*(x_k) = \kappa(x_k) - \kappa(\bar{x}_k)$, that is, there are at least as many positive literals x_k in F as negative literals $\bar{x}_k$. Then we choose a set of $s = \kappa^*(x_k)$ dummy edges $f_{l_1}, \ldots, f_{l_s}$ that have not been included in any pairs yet. We define the following edge pairs, where $t = q - r$: $\{e_{i_h,j_h}, e_{i_{h+r},j_{h+r}}\}$, for $1 \le h \le t$; $\{e_{i_{h+r},j_{h+r}}, e_{i_{h+1},j_{h+1}}\}$, for $1 \le h \le \max(t, r-1)$; $\{e_{i_{h+t},j_{h+t}}, f_{l_h}\}$, for $1 \le h \le s$; and $\{f_{l_h}, e_{i_{h+t+1},j_{h+t+1}}\}$, for $1 \le h < s$. Intuitively, we construct a "path" of edge pairs as shown in Figure 3 where edges corresponding to positive and negative literals alternate until we run out of negative literals; once this happens, we place a dummy edge between every pair of consecutive positive literals. This ensures that all edges corresponding to literal x_k are either all present or all absent; the former is the case if and only if all edges corresponding to literal $\bar{x}_k$ are absent; the latter is the case if and only if all edges corresponding to literal $\bar{x}_k$ are present. Note that, by creating κ^* dummy edges, we ensure that we have enough dummy edges to complete this construction for

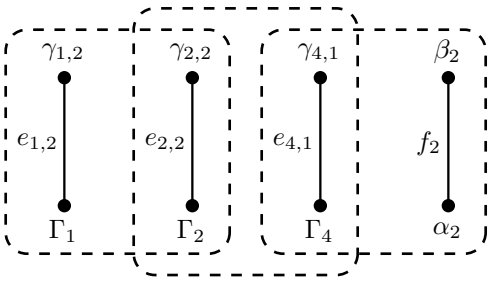

Figure 3: The "path" of alternating literal edges for the variable x_2 in the formula $F = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \wedge (x_1 \vee x_3 \vee \bar{x}_4) \wedge (x_2 \vee x_3 \vee x_4)$. Edge pairs are shown as dotted boxes.

all variables $x_1, \ldots, x_n$ while using every dummy edge in the creation of edge pairs for exactly one variable x_k. This guarantees that, indeed, every edge is in at most two edge pairs, as illustrated in Figure 2.

Now we observe that the vertices in the chain formed by the permanent edges in $G'(F)$ belong to the same connected component H in any graph in $\mathcal{G}(F)$. In a graph $G \in \mathcal{G}(F)$, vertex Γ_i is connected to a vertex in H if and only if the truth assignment corresponding to G satisfies clause C_i. Hence, there is a connected graph in $\mathcal{G}(F)$ if and only if F is satisfiable. Since the construction of $\mathcal{G}(F)$ from F can easily be carried out in linear time, we obtain the following result.

Theorem 1 *Planar k-MCP is NP-hard, for any $k \geq 2$.*

3 Existence of Stable and Merging Flips and Flip Sequences

Given the NP-hardness of k-MCP for $k > 1$, we restrict our attention to 1-MCP in the remainder of this paper. In order to solve this problem, we start with an arbitrary graph G in the given 1-thick family $\mathcal{G}(V, \mathcal{E})$ and then compute a maximizing sequence of flips. In this section, we prove that every graph in $\mathcal{G}(V, \mathcal{E})$ has a maximizing sequence of at most $n - 1$ flips. In the next section, we develop an $\mathcal{O}(n^2 m)$-time algorithm that finds such a sequence. First we prove that every non-maximal graph G in $\mathcal{G}(V, \mathcal{E})$ has a merging flip or a stable flip that leaves the connected components of G invariant, that is, does not change their vertex sets. We call such a stable flip *strongly stable*.

Lemma 1 *Every non-maximal graph G in a family $\mathcal{G}(V, E)$ has a merging or strongly stable flip.*

Proof. Every non-maximal graph contains a cycle. Flipping any edge e in the cycle cannot destroy connected components, but may merge two components if the endpoints of $\bar{e}$ are in different connected components. □

Given a graph G, we call a flip of an edge $e \in G$ *greedy* if the endpoints of edge $\bar{e}$ are in different connected components of G. A sequence $e_1, \ldots, e_q$ of edge flips is greedy if, for every $1 \leq i \leq q$, the flip of edge e_i is greedy for $G\langle e_1, \ldots, e_{i-1}\rangle$. The following two observations establish two important properties of greedy flips.

Observation 1 *The flip of an edge $e \in G$ is merging if and only if it is greedy and e is not a cut edge of G.*

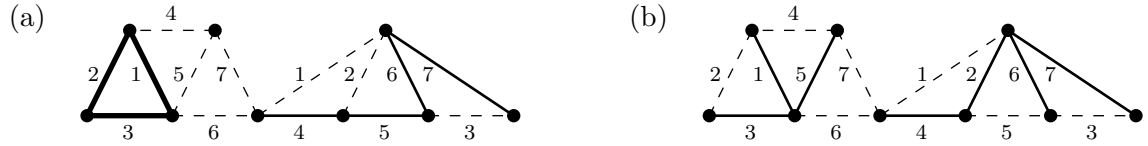

Figure 4: (a) A graph G with no merging flip. Solid edges are in the graph; dotted edges are not in the graph and can be exchanged for edges in the graph through flips. (b) A graph in the same family with one less connected component than G.

Observation 2 *An edge $e \in G$ such that the flip of edge e is greedy and stable is a cut edge of G.*

Another way to interpret Observation 2 is that stable greedy flips leave all cycles in G intact. Also observe that, for every greedy flip e, edge $\bar{e}$ is a cut edge of $G\langle e\rangle$. We are now ready to prove that not every non-maximal graph has a merging flip; but every non-maximal graph has a merging or, in fact, maximizing sequence of at most $n-1$ flips.

Lemma 2 *Not every non-maximal graph has a merging flip.*

Proof. The graph G in Figure 4a is not maximal because the graph in Figure 4b belongs to the same family and has one less connected component. By Observation 1, any merging flip would have to remove an edge in the bold triangle. However, all these flips are stable. □

Lemma 3 *Every non-maximal graph has a maximizing sequence of of most $n-1$ flips.*

Proof. Let G be a non-maximal graph in a 1-thick family $\mathcal{G}(V,\mathcal{E})$, let G^* be a maximal graph in $\mathcal{G}(V,\mathcal{E})$, and let T^* be a spanning forest of G^* that contains a spanning tree for every connected component of G^*. We prove that there exists a maximizing sequence for G that flips at most all the edges in G whose complements are in T^*. Since there are at most $n-1$ such edges, this proves the lemma. The proof is by induction on the number of edges in G whose complements are in T^*. If there is exactly one such edge e, the graph $G\langle e\rangle$ has T^* as a subgraph and is, hence, maximal. Thus, the single flip of edge e is maximizing. So assume that G contains $r > 1$ edges whose complements are in T^* and that the lemma holds for every graph G' that contains less than r edges whose complements are in T^*. Since T^* has fewer connected components than G, there has to be an edge $\bar{e}$ in T^* that connects two vertices in different connected components of G. The flip of edge e is greedy for G; that is, this flip is either merging or stable for G. If $G\langle e\rangle$ is maximal, then the sequence that flips only edge e is maximizing. Otherwise, $G\langle e\rangle$ is a non-maximal graph that contains $r-1$ edges whose complements are in T^*. Hence, by the inductive hypothesis, there exists a maximizing sequence $e_1,\ldots,e_t$ for $G\langle e\rangle$ with $t \leq r-1$. The sequence $e, e_1,\ldots,e_t$ is maximizing for G and has length at most r. □

4 Finding Merging Sequences of Flips

Given that every graph G has a maximizing sequence of at most $n-1$ flips, we would like to compute such a sequence efficiently. Given G and a maximal graph G^*, the construction in the proof of Lemma 3 can easily be implemented in $\mathcal{O}(nm)$ time. The problem is finding G^*. In this section, we provide an $\mathcal{O}(nm)$-time algorithm that finds a merging sequence of at most $n-1$ flips

for any non-maximal graph. By applying this procedure at most $n-2$ times, which takes $\mathcal{O}(n^2m)$ time, we obtain a maximal graph G^*.

Given a graph $G \in \mathcal{G}(V, \mathcal{E})$, we use an auxiliary directed graph H, which is derived from G, to find a merging flip sequence for G. We prove that every shortest path between certain vertices in H has length at most $n-1$ and corresponds to a merging flip sequence and that at least one such path exists if G is non-maximal. Hence, all we have to do is apply breadth-first search to H to either find such a shortest path and report the corresponding flip sequence or output that G is maximal if no such path exists.

The vertex set of H is the edge set of G. For two edges e and f of G, there is an edge (e, f) (directed from e to f) in H if f is a cut edge of G, but not of $G \cup \{\bar{e}\}$. Every edge in G that is not a cut edge corresponds to a source (vertex of in-degree 0) in H; we call such a source a *root*. Every edge e in G such that $\bar{e}$ has its endpoints in different connected components of G corresponds to a sink (vertex of out-degree 0) in H; we call such a sink a *leaf*. Every merging sequence of flips has to flip an edge corresponding to a root, because it has to break at least one cycle in G. It also has to flip at least one edge corresponding to a leaf in H because, otherwise, the flips in the sequence cannot reduce the number of connected components. In particular, if we stop the construction in the proof of Lemma 3 as soon as the number of connected components has reduced by one, we obtain a merging sequence of length at most $n-1$ that starts with a greedy flip, which corresponds to a leaf in H, and ends with a merging flip, which, by Observation 1, corresponds to a root. Our goal is to show that graph H contains a root-to-leaf path if G is not maximal and that every shortest such path corresponds to a merging sequence of flips. So assume that G is not maximal. We call a greedy flip sequence $e_1, \ldots, e_q$ *monotone* if edges $e_1, \ldots, e_q$ are in G.[1] As shown in the proof of Lemma 3, a monotone merging sequence of length at most $n-1$ exists if G is not maximal. The following two lemmas are the first steps toward showing that there exists a root-to-leaf path in H.

Lemma 4 *In a shortest monotone merging sequence $e_1, \ldots, e_q$ for G, edges $e_1, \ldots, e_{q-1}$ are cut edges of G.*

Proof. Assume the contrary and choose j minimal so that e_j is not a cut edge; that is, $e_1, \ldots, e_{j-1}$ are cut edges. We claim that $e_1, \ldots, e_j$ is a monotone merging sequence of flips. This would contradict the assumption that $e_1, \ldots, e_q$ is the shortest such sequence for G. Sequence $e_1, \ldots, e_j$ is certainly monotone, as every subsequence $e_1, \ldots, e_h$ of a monotone sequence of flips must itself be monotone. To see that sequence $e_1, \ldots, e_j$ is merging, we make the following observations: (1) $\omega(G\langle e_1, \ldots, e_{j-1}\rangle) \leq \omega(G)$, because the sequence $e_1, \ldots, e_q$ is merging. (2) Edge $\bar{e}_j$ has its endpoints in different connected components of $G\langle e_1, \ldots, e_{j-1}\rangle$, because the sequence $e_1, \ldots, e_j$ is greedy. (3) The cycles of G are invariant under deletion of cut edges. Hence, e_j is not a cut edge of $G\langle e_1, \ldots, e_{j-1}\rangle$ and, by Observation 1, $\omega(G\langle e_1, \ldots, e_j\rangle) < \omega(G\langle e_1, \ldots, e_{j-1}\rangle) \leq \omega(G)$. □

Lemma 5 *In a shortest monotone merging sequence $e_1, \ldots, e_q$ for G, e_q is not a cut edge of G.*

Proof. Since sequence $e_1, \ldots, e_q$ is a shortest monotone merging sequence, no subsequence $e_1, \ldots, e_j$, $j < q$, is merging. Hence, the flip of edge e_q is merging for $G\langle e_1, \ldots, e_{q-1}\rangle$. By Observation 1, this implies that e_q is not a cut edge of $G\langle e_1, \ldots, e_{q-1}\rangle$. If e_q is a cut edge of G, we choose j minimal so that e_q is not a cut edge of $G\langle e_1, \ldots, e_j\rangle$. Since e_q is a cut edge of $G\langle e_1, \ldots, e_{j-1}\rangle$, the insertion of

[1] A sequence of greedy flips may flip an edge and later flip it back; this is what we disallow here.

edge $\bar{e}_j$ must create a cycle in $G\langle e_1, \ldots, e_j\rangle$. But then the endpoints of $\bar{e}_j$ are in the same connected component of $G\langle e_1, \ldots, e_{j-1}\rangle$, contradicting the greediness of sequence $e_1, \ldots, e_q$. □

Lemma 5 implies that e_q is a root in H. Since the endpoints of edge e_1 are in different connected components of G, by the greediness of sequence $e_1, \ldots, e_q$, edge e_1 is a leaf of H. Hence, to prove that graph H contains a root-to-leaf path of length at most $n-1$ if G is not maximal, it suffices to show that there exists a path from e_q to e_1. We are unable to show exactly this; but we can show that there exists a path of this length from e_q to e_1 or to another leaf of H.

Lemma 6 *If G is not maximal, then there exists a root-to-leaf path of length at most $n-1$ in H.*

Proof. Consider a shortest monotone merging flip sequence $e_1, \ldots, e_q$. Then $q \leq n-1$. We have just observed that e_1 is a leaf and e_q is a root of H. We show that for every edge e_i, $1 \leq i \leq q$, there exists a path of length at most i from e_i to a leaf in H. Hence, there is a path of length at most $n-1$ from e_q to a leaf; e_q is a root. The proof is by induction on i. Since e_1 is itself a leaf, the claim holds for e_1. So assume that $i > 1$ and that the claim holds for $e_1, \ldots, e_{i-1}$. If e_i is a leaf, the claim holds for e_i. Otherwise, the endpoints of $\bar{e}_i$ are in the same connected component of G. But they are in different connected components of $G\langle e_1, \ldots, e_{i-1}\rangle$; so there must be an edge e_j, $j < i$, that is on all paths in G connecting the endpoints of $\bar{e}_i$, because edges $e_1, \ldots, e_{q-1}$ are cut edges. This implies that e_j is an out-neighbour of e_i in H. By the induction hypothesis, there exists a path of length at most j from e_j to a leaf. This implies that there exists a path of length at most $j+1 \leq i$ from e_i to a leaf. □

Given that graph H is guaranteed to contain a root-to-leaf path if G is not maximal, it is natural to ask what the relationship between root-to-leaf paths in H and merging flip sequences for G is. It is easy to show that not every root-to-leaf path in H corresponds to a merging flip sequence. (See Figure 8 in the appendix.) Next we prove that every *shortest* root-to-leaf path in H corresponds to a merging sequence. To prove this fact, we make use of the following two results.

Lemma 7 *For a shortest root-to-leaf path $(e_1, \ldots, e_q)$ in H and any $1 \leq i < q$, edges $\bar{e}_i$ and e_{i+1} are in a common simple cycle of the graph $G\langle e_1, \ldots, e_i\rangle$.*

Proof. Assume that the lemma does not hold. Then let i be minimal so that edges $\bar{e}_i$ and e_{i+1} are not in a common simple cycle in $G\langle e_1, \ldots, e_i\rangle$. Since edge (e_i, e_{i+1}) exists in H, edges $\bar{e}_i$ and e_{i+1} are in a common cycle in $G \cup \{\bar{e}_i\}$. We choose j maximal so that $\bar{e}_i$ and e_{i+1} are in a common cycle of $G\langle e_1, \ldots, e_h\rangle \cup \{\bar{e}_i\}$, for all $0 \leq h \leq j$. Then $j < i$, and edges $\bar{e}_i$ and e_{i+1} do not belong to a common cycle in the graph $G\langle e_1, \ldots, e_{j+1}\rangle \cup \{\bar{e}_i\}$. Hence, edge e_{j+1} is on the cycle in $G\langle e_1, \ldots, e_j\rangle \cup \{\bar{e}_i\}$ that contains edges e_{i+1} and $\bar{e}_i$ (see Figure 5a). Since $\bar{e}_j$ and e_{j+1} belong to the same cycle in $G\langle e_1, \ldots, e_j\rangle$, the removal of edge e_{j+1} still leaves a cycle that contains $\bar{e}_i$ and e_{i+1} (Figure 5b), unless the cycle containing $\bar{e}_j$ and e_{j+1} also contains e_{i+1} (Figure 5c). In the former case, we obtain a contradiction to the assumption that no cycle containing $\bar{e}_i$ and e_{i+1} exists in $G\langle e_1, \ldots, e_{j+1}\rangle \cup \{\bar{e}_i\}$. We prove that, in the latter case, there exists a shorter path from e_1 to e_q in H, which contradicts the assumption that path $e_1, \ldots, e_q$ is a shortest root-to-leaf path in H.

Since edge (e_j, e_{j+1}) exists in H, edge e_{j+1} is a cut edge on the path in G that connects the two endpoints of edge $\bar{e}_j$. If e_{i+1} is also on this path, then edge (e_j, e_{i+1}) exists in H and $(e_1, \ldots, e_j, e_{i+1}, \ldots, e_q)$ is a shorter path from e_1 to e_q in H, a contradiction. So assume that e_{i+1} is not on this path. Edge e_{i+1} is a cut edge of G and both endpoints of edge $\bar{e}_j$ are in the same

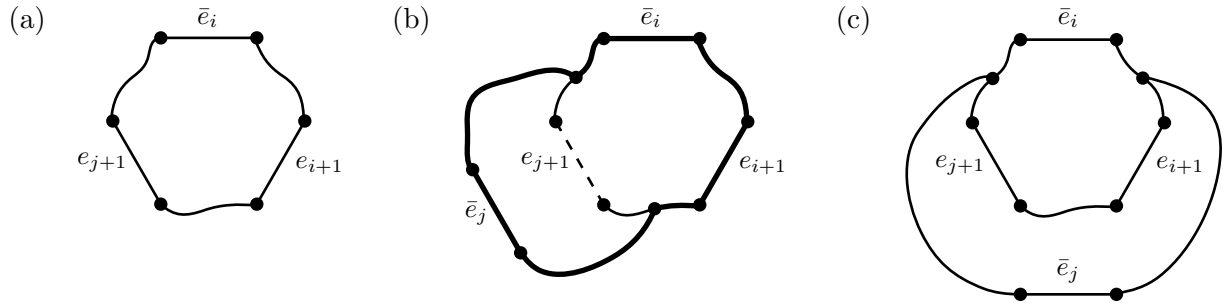

Figure 5: (a) Edges $\bar{e}_i$, e_{i+1}, and e_{j+1} belong to a cycle in $G\langle e_1, \ldots, e_j\rangle \cup \{\bar{e}_i\}$. (b) If edge e_{i+1} is not on the cycle in $G\langle e_1, \ldots, e_j\rangle$ containing edges e_{j+1} and $\bar{e}_j$, then there exists a cycle (bold) in $G\langle e_1, \ldots, e_{j+1}\rangle \cup \{\bar{e}_i\}$ that contains edges $\bar{e}_i$ and e_{i+1}. (c) The case when edge e_{i+1} is on the cycle in $G\langle e_1, \ldots, e_j\rangle$ containing edges e_{j+1} and $\bar{e}_j$.

connected component of $G - e_{i+1}$. Moreover, none of the edges $\bar{e}_1, \ldots, \bar{e}_j$ connects two vertices in different connected components of G. Hence, the only way to create a path in $G\langle e_1, \ldots, e_{j-1}\rangle$ that connects the endpoints of edge $\bar{e}_j$ and contains edge e_{i+1} is by adding an edge $\bar{e}_h$, $h < j$, whose endpoints are in different connected components of $G - e_{i+1}$, but in the same connected component of G (see Figure 6). Then, however, edge (e_h, e_{i+1}) exists in H and the path $(e_1, \ldots, e_h, e_{i+1}, \ldots, e_q)$ is a shorter path from e_1 to e_q in H, again a contradiction. □

Corollary 1 *Let $e_1, \ldots, e_q$ be a shortest root-to-leaf path in H. Then, for every $1 \leq i < q$, the flip of edge e_i is strongly stable for $G\langle e_1, \ldots, e_{i-1}\rangle$.*

Using Lemma 7 and Corollary 1, we can now prove that every shortest root-to-leaf path in H corresponds to a merging sequence of flips for G.

Lemma 8 *A shortest root-to-leaf path in H corresponds to a merging sequence of flips for G.*

Proof. Consider a shortest root-to-leaf path $e_1, \ldots, e_q$ in H. By Corollary 1, all flips in the sequence $e_1, \ldots, e_{q-1}$ are strongly stable. By Lemma 7, edge e_q belongs to a cycle in $G\langle e_1, \ldots, e_{q-1}\rangle$. By Corollary 1, edge $\bar{e}_q$ has its endpoints in different connected components of $G\langle e_1, \ldots, e_{q-1}\rangle$, because this is true in G. Hence, by Observation 1, the flip of edge e_q is merging for $G\langle e_1, \ldots, e_{q-1}\rangle$ and the whole sequence $e_1, \ldots, e_q$ is merging for G. □

Given this correspondence between shortest root-to-leaf paths in H and merging sequences for G, we can find a merging sequence of flips for G in $\mathcal{O}(nm)$ time: First we create the vertex set of graph H by adding a vertex for every edge of G. Next we identify the cut edges of G and label all those vertices in H as roots whose corresponding edges in G are not cut edges. We contract every 2-edge connected component into a single vertex and call the resulting graph G'. We compute its connected components, which are trees, and root each such tree at an arbitrary vertex. To identify the edge set of H and the leaves of H, we scan the set of complementary edges of the edges in G. We discard edges that have become loops as the result of the contraction of the 2-edge connected components of G, because they run parallel to a path in a 2-edge connected component of G and, hence, neither have any out-neighbours nor are leaves in H. We mark a vertex e in H as a leaf if

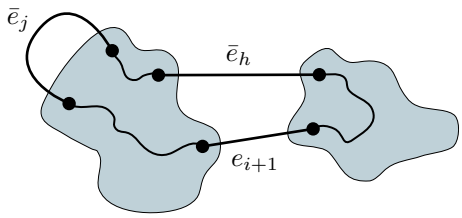

Figure 6: The proof that there must be an edge $(\bar{e}_h, e_{i+1})$ in H, where $h < i$, if edge e_{i+1} is on the cycle in $G\langle e_1, \ldots, e_j\rangle$ that contains edges e_{j+1} and $\bar{e}_j$.

the endpoints of edge $\bar{e}$ are in different connected components of G'. For every edge $\bar{e}$ that is not a loop and that has its endpoints in the same connected component of G', we add an edge (e, f) to H, for every edge f on the tree path connecting the endpoints of $\bar{e}$. These edges can be identified by traversing paths from the endpoints of $\bar{e}$ to their LCA in the tree. Since there are at most $n-1$ cut edges in G, the edge set of G' has size at most $n-1$; the vertex set has size at most n. There are m edges $\bar{e}$. Hence, after constructing G', which takes $\mathcal{O}(n+m)$ time, it takes $\mathcal{O}(nm)$ time to construct H. Now we decide whether there exists a root-to-leaf path in H and, if so, find a shortest such path by running BFS simultaneously from all roots; that is, we place all the roots at the first level of the BFS and then grow the BFS-forest as usual level by level. This takes $\mathcal{O}(nm)$ time. If G is not maximal, our discussion implies that this procedure finds a root-to-leaf path. Since we use BFS to find it, it is a shortest such path; in particular, the resulting path has length at most $n-1$. Hence, we report the sequence of flips corresponding to the vertices on the path as a merging sequence of flips.

Theorem 2 *It takes $\mathcal{O}(nm)$ time to decide whether a graph $G \in \mathcal{G}(V, \mathcal{E})$ is maximal and, if not, find a merging sequence of at most $n-1$ flips.*

Since any graph in $\mathcal{G}(V, \mathcal{E})$ has at most $n-1$ connected components, unless $\mathcal{E}$ is empty, we can apply the algorithm sketched above at most $n-2$ times before we obtain a connected graph, which is obviously maximal. Hence, we can start with an arbitrary graph G in $\mathcal{G}(V, \mathcal{E})$ and repeatedly apply Theorem 2 to compute a maximizing sequence of at most $(n-2)(n-1)$ flips. We apply these flips in $\mathcal{O}(n^2)$ time to obtain a maximal graph $G^* \in \mathcal{G}(V, \mathcal{E})$. As pointed out at the beginning of this section, we can compute, in $\mathcal{O}(nm)$ time, a maximizing sequence of at most $n-1$ flips for G, once G^* is given. This proves the following result.

Corollary 2 *It takes $\mathcal{O}(n^2 m)$ time to compute a maximal graph in $\mathcal{G}(V, \mathcal{E})$ and to compute a maximizing sequence of at most $n-1$ flips for a given graph G in $\mathcal{G}(V, \mathcal{E})$.*

5 Connectivity of Sub-Families Under Edge Flips

Since every graph in a 1-thick family $\mathcal{G}(V, \mathcal{E})$ can be transformed into a maximal graph, an interesting question to ask is whether for any two graphs G_1 and G_2 in $\mathcal{G}(V, \mathcal{E})$ with at most k connected components, there exists a flip sequence $e_1, \ldots, e_q$ that transforms G_1 into G_2 and such that $\omega(G_1\langle e_1, \ldots, e_i\rangle) \leq k$, for all $1 \leq i \leq q$. We call such a sequence *k-stable*. Note that some of the flips in a k-stable sequence may be splitting. Another question to ask is how many flips such a sequence has to contain. Formally, we ask whether the graph $\mathcal{G}(V, \mathcal{E}, k)$ is connected and

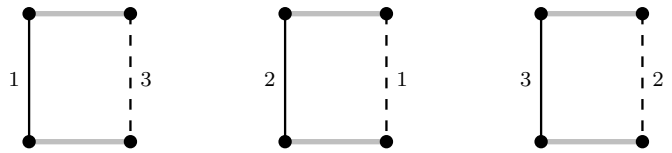

Figure 7: A family $\mathcal{G}(V,\mathcal{E})$ of graphs such that $\mathcal{G}(V,\mathcal{E},\tilde{\omega}(\mathcal{G}(V,\mathcal{E})))$ is disconnected.

what its diameter is. We prove that $\mathcal{G}(V,\mathcal{E},k)$ is connected, for every $k > \tilde{\omega}(\mathcal{G}(V,\mathcal{E}))$, and that $\mathcal{G}(V,\mathcal{E},\tilde{\omega}(\mathcal{G}(V,\mathcal{E})))$ may be disconnected. We also show that, for $k > \tilde{\omega}(\mathcal{G}(V,\mathcal{E}))$, $\mathcal{G}(V,\mathcal{E},k)$ has diameter at most $m+n-1$. To show this, we prove a fact that is in a sense orthogonal to Lemma 3: While Lemma 3 shows that every non-maximal graph can be transformed into *some* maximal graph using a *merging* sequence of at most $n-1$ flips, we prove next that every non-maximal graph can be transformed into *any* maximal graph using a *k-stable* sequence of at most m flips, some of whose flips may be splitting.

Lemma 9 *For any graph family $\mathcal{G}(V,\mathcal{E})$ and any $k > \tilde{\omega}(\mathcal{G}(V,\mathcal{E}))$, $\mathcal{G}(V,\mathcal{E},k)$ is connected and has diameter at most $m+n-1$.*

Proof sketch. The proof of Lemma 3 can easily be adapted to show that any non-maximal graph G_1 can be transformed into any maximal graph G_2 using a k-stable sequence of at most m flips. Indeed, the only difference is that, instead of stopping when a maximal graph G' is obtained, we keep flipping edges until G_2 is obtained. The crucial observation is that, if the current graph G' is not maximal, there exists an edge in G' that is not in G_2 and whose flip is merging. If G' is maximal, we can flip any edge e in G' that is not in G_2; this will result in a graph $G'\langle e\rangle$ with $\omega(G'\langle e\rangle) \leq \omega(G') + 1 \leq k$.

The lemma now follows because, for any two graphs G_1 and G_2 in $\mathcal{G}(V,\mathcal{E},k)$, we first find a maximizing sequence $e_1,\ldots,e_q$ of of at most $n-1$ flips that transforms G_1 into a maximal graph G_3. Such a sequence exists by Lemma 3. Then we find a k-stable sequence $e'_1,\ldots,e'_r$ of at most m flips that transforms G_2 into G_3. Sequence $e_1,\ldots,e_q,\bar{e}'_r,\ldots,\bar{e}'_1$, which has length $q+r \leq m+n-1$, transforms G_1 into G_2 and is k-stable. □

The statement of Lemma 9 is true for any $k > \tilde{\omega}(\mathcal{G}(V,\mathcal{E}))$. For $k = \tilde{\omega}(\mathcal{G}(V,\mathcal{E}))$, the example in Figure 7 shows that $\mathcal{G}(V,\mathcal{E},\tilde{\omega}(\mathcal{G}(V,\mathcal{E})))$ is not necessarily connected: The horizontal edges are permanent; that is, they represent connector graphs. Two graphs in the family are G_1, which includes the permanent edges and the solid vertical edges, and G_2, which includes the permanent edges and the dashed vertical edges. Flipping any vertical edge in G_1 increases the number of connected components by one. Hence, there is no $\tilde{\omega}(\mathcal{G}(V,\mathcal{E}))$-stable sequence of flips that transforms G_1 into G_2, which proves the following lemma.

Lemma 10 *There exists a graph family $\mathcal{G}(V,\mathcal{E})$ such that $\mathcal{G}(V,\mathcal{E},\tilde{\omega}(\mathcal{G}(V,\mathcal{E})))$ is disconnected.*

6 Open Problems

The algorithm for finding a merging sequence of flips takes $\mathcal{O}(nm)$ time. We do believe that there exists an $\mathcal{O}(n+m)$-time algorithm for this problem. If this is true, we would be able to compute a maximal graph in a 1-thick graph family $\mathcal{G}(V,\mathcal{E})$ in $\mathcal{O}(nm)$ time. What is a lower bound for the

running time of any algorithm that solves 1-MCP? Can one obtain a more efficient algorithm for planar 1-MCP?

One can ask similar questions about the behaviour of the biconnected components of a graph under edge flips. We define the maximal biconnectivity problem (MBP) as the problem of finding a graph in a family $\mathcal{G}(V, \mathcal{E})$ that has the minimal number of biconnected components. We ask whether (planar) k-MBP is NP-hard for $k \geq 2$. We believe that the answer to this question is "yes" and that some adaptation of our proof from Section 2 proves this. The more interesting question is whether and how our algorithm for 1-MCP can be adapted to solve 1-MBP efficiently.

Acknowledgements. I would like to thank Anil Maheshwari for some insightful discussion on the subject, Ferran Hurtado for giving a talk that made me see 1-MCP as an edge-flip problem, and Herbert Edelsbrunner for pointing out a possible improvement of Lemma 3.

References

1. O. Aichholzer, F. Aurenhammer, P. Brass, and H. Krasser. Pseudo-triangulations from surfaces and a novel type of edge flip. *SIAM Journal on Computing*, 32(6):1621–1653, 2003.
2. O. Aichholzer, F. Aurenhammer, and F. Hurtado. Sequences of spanning trees and a fixed tree theorem. *Computational Geometry: Theory and Applications*, 21(1–2):3–20, 2002.
3. O. Aichholzer, F. Aurenhammer, and H. Krasser. Adapting (pseudo)-triangulations with a near-linear number of edge flips. In *Proceedings of the 8th International Workshop on Algorithms and Data Structures*, volume 2748 of *Lectures Notes in Computer Science*, pages 12–24. Springer-Verlag, 2003.
4. B. Delaunay. Sur la sphère vide. *Izvestiya Akademii Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7:793–800, 1934.
5. H. Edelsbrunner. Personal communication. 2003.
6. H. Edelsbrunner and D. V. Nekhayev. Repairing self-intersections of triangulated surfaces in space. Technical Report rgi-tech-03-053, Raindrop Geomagic Inc., 2003.
7. H. Edelsbrunner and N. R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15:223–241, 1996.
8. S. Fortune. Voronoi diagrams and Delaunay triangulations. In *Computing in Euclidean Geometry*, D. Z. Hu and F. K. Wang, (eds.), pages 225–265. World Scientific, Singapore, 2nd edition, 1995.
9. J. Galtier, F. Hurtado, M. Noy, S. Pérennes, and J. Urrutia. Simultaneous edge flipping in triangulations. *International Journal on Computational Geometry and Applications*, 13(2):113–133, 2003.
10. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, 1979.
11. L. J. Guibas, D. E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 7:381–413, 1992.
12. F. Hurtado, M. Noy, and J. Urrutia. Flipping edges in triangulations. *Discrete and Computational Geometry*, 22:333–346, 1999.
13. B. Joe. Three-dimensional triangulations from local transformations. *SIAM Journal on Scientific and Statistical Computing*, 10:718–741, 1989.
14. B. Joe. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8:123–142, 1991.
15. S. Negami. Diagonal flips of triangulations on surfaces, a survey. *Yokohama Mathematical Journal*, 47:1–40, 1999.
16. V. T. Rajan. Optimality of the delaunay triangulation in $\Re^d$. *Discrete & Computational Geometry*, 12:189–202, 1994.

Appendix

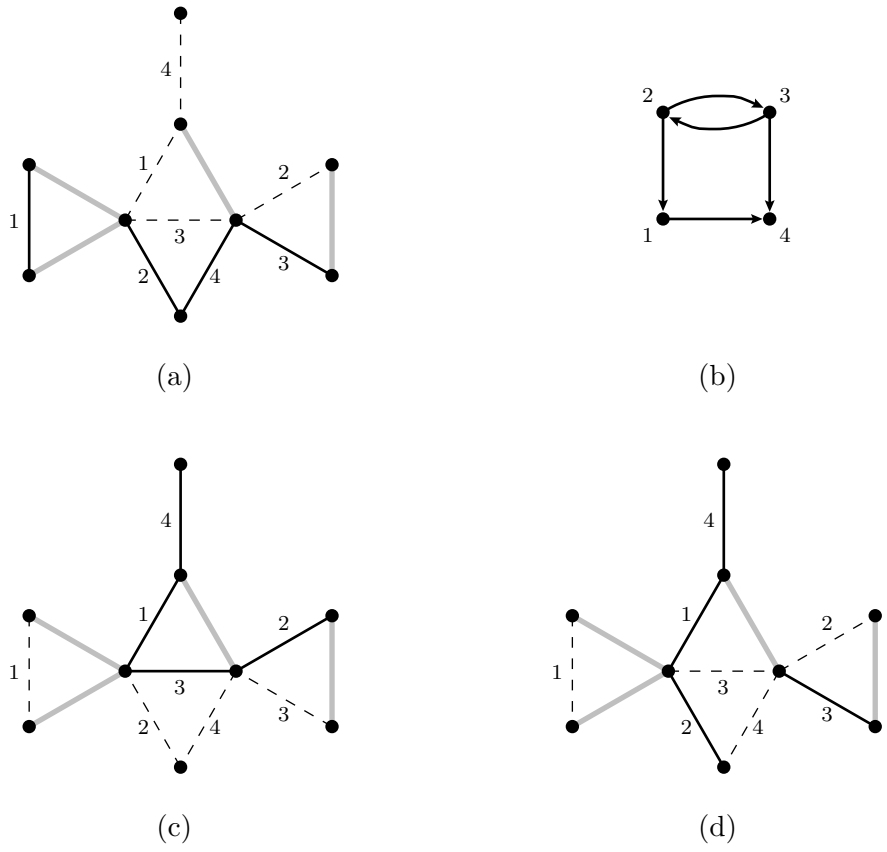

Figure 8: (a) A graph G. (b) Its auxiliary graph H. (c) The graph $G\langle e_1, e_2, e_3, e_4\rangle$ has the same number of connected components as G, even though (e_1, e_2, e_3, e_4) is a root-to-leaf path in H. (d) The graph $G\langle e_1, e_4\rangle$ has one less connected component than G; the path (e_1, e_4) is a shortest root-to-leaf path in H.