Augmenting the edge connectivity of planar straight line graphs to three

Marwan Al-Jubeh* Mashhood Ishaque* Kristóf Rédei* Diane L. Souvaine*
Csaba D. Tóth*† Pavel Valtr‡

Abstract

We characterize the planar straight line graphs (PSLGs) that can be augmented to 3-connected and 3-edge-connected PSLGs, respectively. We show that if a PSLG with n vertices can be augmented to a 3-edge-connected PSLG, then at most $2n-2$ new edges are always sufficient and sometimes necessary for the augmentation. If the input PSLG is, in addition, already 2-edge-connected, then $n-2$ new edges are always sufficient and sometimes necessary for the augmentation to a 3-edge-connected PSLG.

1 Introduction

Connectivity augmentation is a classical problem in graph theory with application in network design. Given a graph $G(V, E)$, with vertex set V and edge set E, and a constant $k \in \mathbb{N}$, find a minimum set E' of *new edges* such that $G'(V, E \cup E')$ is k-connected (respectively, k-edge-connected). Eswaran and Tarjan [4] and Plesník [29] showed independently that the 2-edge-connectivity augmentation problem can be solved in linear time. They also gave a polynomial time solution for the 2-connectivity problem, which was later improved to linear time [34, 15, 16]. Watanabe and Nakamura [40] proved that the edge-connectivity augmentation problem can be solved in polynomial time for every $k \in \mathbb{N}$. The runtime was later improved (using the edge-splitting technique of Lovász [22] and Mader [23]) by Frank [6] and by Nagamochi and Ibaraki [26]. Jackson and Jordán [17] proved that the vertex-connectivity augmentation problem can be solved in polynomial time for every $k \in \mathbb{N}$. Very recently, Végh [39] gave a polynomial time algorithm for the k-connectivity augmentation of a $(k-1)$-connected graph for any k (where k is part of the input). For related problems, refer to surveys by Nagamochi and Ibaraki [27], and by Kortsarz and Nutov [20].

The results on connectivity augmentation of *abstract* graphs do not apply if the input is given with a planar embedding which has to be respected by the new edges (e.g., in case of physical communication or transportation networks). A *planar straight line graph* (PSLG) is a graph $G = (V, E)$, where V is a set of distinct points in the plane, and E is a set of straight line segments between the points in V such that two segments may intersect at their endpoints only. Given a PSLG $G = (V, E)$ and an integer $k \in \mathbb{N}$, the *embedding preserving k-connectivity* (resp., *k-edge-connectivity*) *augmentation* problem asks for a set of new edges E' of minimal cardinality such that $G = (V, E \cup E')$ is a k-connected (resp., k-edge-connected)

*Department of Computer Science, Tufts University, Medford, MA, USA, Email: {maljub01, mishaq01, kredei01, dls, cdtoth}@cs.tufts.edu Partially supported by NSF grant CCF-0830734.

†Department of Mathematics, University of Calgary, Calgary, AB, Canada, Email: cdtoth@ucalgary.ca Research by Tóth was conducted while visiting the Rényi Institute, Budapest, and Tufts University, Medford, MA. Supported in part by NSERC grant RGPIN 35586.

‡Department of Applied Mathematics and Institute for Theoretical Computer Science, Charles University, Prague, Czech Republic. Research by P. Valtr was supported by the projects 1M0545 and MSM0021620838 of the Ministry of Education of the Czech Republic.

PSLG. Since no planar graph is 6-connected or 6-edge-connected, the embedding preserving k-connectivity (k-edge-connectivity) augmentation problems make sense for $1 \le k \le 5$ only.

Rutter and Wolff [35] showed that both the embedding preserving vertex- and edge-connectivity augmentation problems are NP-hard for $k = 2, \ldots, 5$. They reduce PLANAR3SAT to a decision problem whether a PSLG with $2m$ vertices of degree $k-1$ can be augmented to a k-edge-connected PSLG with at most m new edges. The preservation of the input embedding imposes a severe restriction: for example, a path (as an abstract graph) can be augmented to a 2-connected graph by adding one new edge—however if a path is embedded as a "zig-zag" path with n vertices in convex position, then it takes $n-2$ (resp., $\lfloor n/2 \rfloor$) new edges to augment it to a 2-connected (resp., 2-edge-connected) PSLG [1]. If the vertices of a PSLG G are in convex position, then it *cannot* be augmented to a 3-connected (resp., 3-edge-connected) PSLG, since any *maximal* augmentation (which is a triangulation) on $n \ge 3$ vertices in convex position has a vertex of degree 2.

In this paper, we approach the connectivity augmentation for PSLGs from an extremal combinatorial perspective, and determine the *minimum* number of new edges sufficient for augmenting *any* PSLG on n vertices. One of our main results states that if an arbitrary PSLG on n vertices can be augmented to a 3-edge-connected PSLG, then at most $2n-2$ new edges are always sufficient and sometimes necessary for the augmentation.

A few previous combinatorial bounds are known on the minimum number of new edges sufficient for the embedding preserving augmentation of PSLGs. It is easy to see that every PSLG with $n \ge 2$ vertices and p connected components can be augmented to a *connected* PSLG by adding at most $p-1 \le n-1$ new edges. It is also known that every connected PSLG G with $n \ge 3$ vertices and $b \ge 2$ distinct 2-connected blocks can be augmented to a *2-connected* PSLG by adding at most $b-1 \le n-2$ new edges [1]. Every connected PSLG with $n \ge 3$ vertices can be augmented to a *2-edge-connected* PSLG by adding at most $\lfloor (2n-2)/3 \rfloor$ new edges [37]. These bounds are tight in the worst case [1].

Results of García et al. [8] imply that every *empty* PSLG on n vertices in general position can be augmented to a 3-edge-connected PSLG with at most $2n-2$ (new) edges, and this bound is tight.

Results. We characterize PSLGs that can be augmented to 3-connected or 3-edge-connected PSLGs, respectively. We say that a PSLG is *3-augmentable* (resp., *3-edge-augmentable*) if it admits an embedding preserving augmentation to a 3-connected (resp., 3-edge-connected) PSLG. We show the following (Section 2):

- A PSLG $G = (V, E)$ is 3-augmentable if and only if V is not in weakly convex position and E does not contain a chord of the convex hull of V;
- a PSLG $G = (V, E)$ is 3-edge-augmentable if and only if V is not in weakly convex position and there is no edge $e \in E$ such that the endpoints of e and all vertices on one side of e lie on the boundary of the convex hull of V.

We also give worst-case tight bounds on the minimum number of new edges that have to be added to a 3-edge-augmentable PSLG to obtain a 3-edge-connected PSLG. We prove the following:

- Every 3-edge-augmentable 2-edge-connected PSLG on n vertices can be augmented to a 3-edge-connected PSLG with at most $n-2$ new edges, and this bound cannot be improved (Theorem 1);
- every 3-edge-augmentable PSLG with n vertices can be augmented to 3-edge-connected PSLG with at most $2n-2$ new edges, and this bound cannot be improved (Theorem 2).

In both cases, we provide explicit algorithms for constructing the augmentation. If the input PSLG with n vertices is already 2-edge-connected, then our augmentation algorithm runs in $O(n\alpha(n))$ time, where $\alpha(n)$ is the inverse Ackermann function. For an arbitrary 3-edge-augmentable PSLG with n vertices in general position, the augmentation algorithm runs in $O(n \log^2 n)$ time. Both runtimes are in the real RAM model of computation (Section 7).

Remarks. A few remarks are in order. For *abstract* graphs (with no planar embedding), it is easy to deduce worst-case bounds on the number of new edges necessary to augment the vertex- or edge-connectivity to three. The empty graph with $n \geq 3$ vertices can be augmented to a Hamiltonian circuit with n new edges. A Hamiltonian circuit on $n \geq 4$ vertices can be augmented to a 3-connected graph with $\lceil n/2 \rceil$ additional edges (by connecting opposite vertices along the circuit). So the empty graph with $n \geq 4$ vertices can be augmented to a 3-connected (hence 3-edge-connected) graph with $\lceil 3n/2 \rceil$ edges. This bound is best possible since the degree of every vertex is at least 3 in a 3-edge-connected (or 3-connected) graph. It follows that *every* graph G with $n \geq 4$ vertices can be augmented to a 3-connected graph by adding at most $\lceil 3n/2 \rceil$ new edges (since we can superimpose G with the above graph constructed on the empty graph), and this bound is worst-case optimal.

In connectivity augmentation for abstract graphs, it is a common technique to augment a j-edge-connected graph to a k-edge-connected graph in $k - j$ stages, where each stage raises the edge-connectivity by one [3, 39]. However, our worst-case bound on PSLGs does not follow from a combination of worst case bounds for augmenting the edge-connectivity by one. In the worst case, we need $n - 1$ new edges to augment the (edge-) connectivity of a PSLG to 1 (which is tight for the empty graph); we need $\lfloor (2n-2)/3 \rfloor$ new edges to augment the edge-connectivity from 1 to 2 [1]; and we need $n - 2$ new edges to augment the edge-connectivity from 2 to 3 (Section 3 below). The naïve combination of the worst-case bounds shows that a PSLG on $n \geq 4$ vertices can be augmented to a 3-edge-connected PSLG by adding at most $(n-1) + \lfloor (2n-2)/3 \rfloor + (n-2) = \lfloor (8n-11)/3 \rfloor$ new edge. We prove, however, that $2n - 2$ new edges are always sufficient and sometimes necessary.

Related work. An augmentation is *planarity preserving* if both G and G' are planar (i.e., do not contain K_5 or $K_{3,3}$ as a minor). A given planar graph cannot be augmented to become a k-connected planar graph for an arbitrarily large $k \in \mathbb{N}$. In particular, planarity preserving augmentation problems do not make sense for $k > 5$, since no planar graph is 6-connected or 6-edge-connected. Kant and Bodlaender [19] showed that the planarity preserving vertex-connectivity augmentation problem is NP-complete already for $k = 2$. Fiala and Mutzel [5] presented an approximation algorithm in $O(n^3)$ time for $k = 2$, the approximation ratio has recently been proved to be 2 [13]. Kant and Bodlaender [19] proposed a $\frac{5}{4}$-approximation in $O(n^3)$ time for $k = 3$. Linear time algorithms for the planarity preserving versions are known for the case that $k = 2$ and the input G is an outerplanar graph [18, 25]; and for the version of the problem where both the input G and the output G' are required to be outerplanar [9].

2 Which graphs can be augmented?

In this section we study 3-connected and 3-edge-connected triangulations, and then we characterize the PSLGs that admit embedding preserving augmentations to 3-connected and 3-edge-connected straight line triangulations, respectively.

For a set V of points in the plane (e.g., a vertex set of a PSLG), let $\mathrm{ch}(V)$ denote the boundary of the convex hull of V. If the points in V are not all collinear, then $\mathrm{ch}(V)$ is a simple polygon. For a PSLG G we use the shorthand notation $\mathrm{ch}(G) := \mathrm{ch}(V(G))$. The point set V is in *weakly convex position* if all points in V lie on $\mathrm{ch}(V)$ (but there may be three or more collinear points along $\mathrm{ch}(V)$). A *chord* of $\mathrm{ch}(G)$ is an edge such that its endpoints lie on $\mathrm{ch}(G)$, and its relative interior lies in the interior of $\mathrm{ch}(G)$. A *triangulation* of a point set V in the plane is a connected PSLG on the vertex set V such that all bounded faces are triangles and the unbounded face is the complement of $\mathrm{ch}(V)$. If V is a set of n non-collinear points, k of which lie in the interior of $\mathrm{ch}(V)$, then every triangulation of V has exactly $n + k - 2$ bounded triangular faces. It is well known that every PSLG can be augmented to a triangulation.

Proposition 2.1 *Let T be a triangulation on $n \geq 4$ non-collinear vertices. Then*

- *T is 3-connected if and only if no edge of T is a chord of $\mathrm{ch}(T)$;*
- *T is 3-edge-connected if and only if no bounded face of T has two edges along $\mathrm{ch}(T)$.*

Proof. If an edge $e \in E(T)$ is a chord of $\mathrm{ch}(T)$, then the removal of the two endpoints of e disconnects T. Assume now that no edge $e \in E(T)$ is a chord of $\mathrm{ch}(T)$. We show that T is 3-connected, that is, it remains connected after the removal of any two vertices (and all incident edges). Consider two arbitrary vertices $u, v \in V$, and suppose w.l.o.g. that segment uv is vertical. From any vertex $w \in V \setminus \{u, v\}$ on or to the left (resp., right) of the supporting line of uv, the triangulation T contains an x-monotone path to a leftmost (resp., rightmost) vertex in V, since all bounded faces are convex. These paths avoid both u and v. Between every leftmost and every rightmost vertex in V, there are two paths along $\mathrm{ch}(T)$, since not all vertices are collinear. At least one of these paths avoids both u and v, otherwise uv would be a chord of the convex hull. Therefore for any two vertices $w_1, w_2 \in V \setminus \{u, v\}$, we can construct a path in T that avoids both u and v by concatenating paths to a leftmost or a rightmost vertex in V, and connecting them along $\mathrm{ch}(T)$. This proves that T is 3-connected.

If a bounded face of a triangulation T has two edges along $\mathrm{ch}(T)$, then the removal of these two edges disconnects T. Assume now that T can be disconnected by removing two edges $e, f \in E(T)$. Then there is a closed curve γ that separates the components of $G \setminus \{e, f\}$ such that γ crosses e and f but no other edges (and does not pass through any vertex). So γ traverses exactly two faces. These faces share two edges, e and f. Since no two triangles share two edges, one of the faces is the outer face, and the other face is a triangle. Since any two edges of a triangle are adjacent, e and f are adjacent along $\mathrm{ch}(T)$. □

Proposition 2.2 *A PSLG $G = (V, E)$ with $n \geq 4$ vertices is 3-augmentable if and only if the vertices of G are not in weakly convex position and no edge of G is a chord of $\mathrm{ch}(G)$.*

Proof. First we show that if V is in weakly convex position or E contains a chord of $\mathrm{ch}(G)$, then G is not 3-augmentable. Assume, to the contrary, that G admits an embedding preserving augmentation to a 3-connected PSLG G'. Since additional edges can only increase the connectivity, we may assume that G' is a triangulation. If V is in weakly convex position, then any triangulation of V contains a vertex of degree two, so G' cannot be 3-connected. If G contains a chord of $\mathrm{ch}(G)$, then so does G', which implies that G' cannot be 3-connected.

Assume now that $G = (V, E)$ is a PSLG such that V is not in weakly convex position and E contains no chord of $\mathrm{ch}(G)$. It is enough to augment G to a 3-connected triangulation T, this already implies that G is 3-augmentable. By Proposition 2.1, T will be 3-connected if none of its edges is a chord of $\mathrm{ch}(G)$. First augment G with all edges along the convex hull $\mathrm{ch}(G)$. Then for every vertex in the interior of $\mathrm{ch}(G)$, greedily add new incident edges as long as they do not cross existing edges. Denote the resulting graph by G'. Note that no edge of G' is a chord of $\mathrm{ch}(G)$, and every bounded face of G' is convex. We show that every bounded face of G' is a triangle, and so G' is the required triangulation. Assume, to the contrary, that G' has a convex face F with at least 4 vertices. All vertices of F are on the convex hull of G, otherwise a vertex of F in the interior of $\mathrm{ch}(G)$ could be connected to another vertex of F (thereby decomposing F into smaller faces). However, the edges of F are not chords, so all edges of F lie along $\mathrm{ch}(G)$. Therefore, all vertices of G' are incident to F, that is, $V(G)$ is in weakly convex position, contradicting our assumption. □

Proposition 2.3 *A* PSLG *G with $n \geq 4$ vertices is 3-edge-augmentable if and only if the vertices of G are not in weakly convex position and no edge e of G has both of the following two properties:*

(i) *e is a chord of $\mathrm{ch}(G)$ and*

(ii) *all vertices of G lying on one side of the supporting line of e lie on $\mathrm{ch}(G)$.*

Proof. First we show that if V is in weakly convex position or E contains a chord of $\mathrm{ch}(G)$ such that all vertices on one of its sides lie on $\mathrm{ch}(G)$, then G is not 3-edge-augmentable. Assume, to the contrary, that G admits an embedding preserving augmentation to a 3-edge-connected PSLG G'. Since additional edges can only increase the connectivity, we may assume that G' is a triangulation. If V is in weakly convex position, then any triangulation of V contains a vertex of degree two, so G' cannot be 3-edge-connected. If G contains a chord uv of $\mathrm{ch}(G)$ such that all vertices on one side of uv lie on $\mathrm{ch}(G)$, then let V_1 be the set of vertices on this side of uv together with the endpoints u and v. The subgraph of T induced by V_1 is a triangulation T_1 on at least three vertices. If $|V_1| = 3$, then the single vertex in $V_1 \setminus \{u, v\}$ has degree two in both T_1 and G'. If $|V_1| \geq 4$, then note that every triangulation on at least 4 vertices contains two nonadjacent vertices of degree two (since the dual graph of the triangulation has at least two leaves). Hence, there is a vertex in $V_1 \setminus \{u, v\}$ of degree two in T_1. The degree of this node is 2 in G' as well, since it cannot be connected to the vertices on the opposite side of uv.

Assume now that $G = (V, E)$ is a PSLG such that V is not in weakly convex position, and E contains no edge satisfying both (i) and (ii). It is sufficient to augment G to a 3-edge-connected triangulation T in which no new edge satisfies both (i) and (ii). Such a triangulation T will not have a triangle with two edges along the convex hull, since the third edge would be a chord such that there is exactly one vertex on one side, which lies on the convex hull. By Proposition 2.1, T will be 3-edge-connected.

First augment G with all edges along the convex hull $\mathrm{ch}(G)$. Then for every vertex in the interior of $\mathrm{ch}(G)$, greedily add new incident edges as long as they do not cross existing edges. Denote the resulting graph by G'. Note that no edge of G' is a chord of $\mathrm{ch}(G)$, and every bounded face of G' is convex. If every bounded face of G' is a triangle, then G' is the required triangulation. Consider a convex face F of G' with at least 4 vertices. All vertices of F are on the convex hull of G, otherwise a vertex of F in the interior of $\mathrm{ch}(G)$ could be connected to another vertex of F (thereby decomposing F into smaller faces). At least two edges of F are chords of $\mathrm{ch}(G)$, since if no edge of F is a chord then V is in weakly convex position, and if exactly one edge of F is a chord, then G has a chord satisfying (i) and (ii). Let e and f be two edges of F that are chords of G. Note that G must have vertices in the interior of $\mathrm{ch}(G)$ on the opposite side of e and f each. Triangulate face F with edges whose supporting lines separate e and f. The new edges are chords such that there are vertices in the interior of $\mathrm{ch}(G)$ on both sides. □

3 Lower bound constructions

We construct PSLGs with n vertices that are 3-edge-augmentable, but cannot be augmented to a 3-edge-connected PSLG with fewer than $2n - 2$ new edges. We present two families of lower bound constructions.

First, consider the empty graph with $n-1$ vertices in convex position, and one vertex in the interior of the convex hull, for $n \geq 4$ (see Fig. 1, left). The only 3-edge-connected augmentation is a wheel graph with $2n-2$ edges. Second, consider a triangulation with m vertices, $2m-5$ bounded faces and the outer face with exactly three edges. Put a singleton vertex in each bounded face, and 2 singletons next to each edge in the outer face as in Fig. 1, right. The only 3-edge-connected augmentation is obtained by adding 3 new edges at each singleton in a bounded face, and 5 new edges for each pair of singletons next to an edge of the outer face. A graph with $n = m + (2m-5) + 6 = 3m + 1$ vertices requires $(2m-5)3 + 3 \cdot 5 = 2n-2$ new edges.

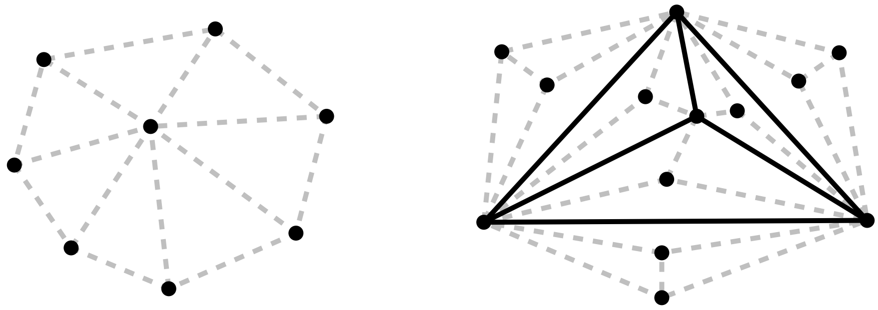

Figure 1: Left: $n-1$ vertices in convex position and one vertex in the interior of the convex hull. Right: a triangulation with $m = 4$ vertices, $2m-5 = 3$ singletons in bounded faces, and 2 singletons next to each edge in the outer face.

If the input graph is already 2-edge-connected, then fewer new edges are enough to obtain a 3-edge-connected PSLG. We present PSLGs that are 3-edge-augmentable and 2-edge-connected, but cannot be augmented to a 3-edge-connected PSLG with fewer than $n-2$ new edges. Consider a Hamiltonian circuit on n vertices with $n-1$ vertices in convex position and one vertex lying in the interior of the convex hull as in Fig. 2. This PSLG has a unique embedding preserving augmentation to a 3-edge-connected PSLG, which is obtained by connecting the vertex in the interior of the convex hull to all $n-3$ nonadjacent vertices, and adding the one missing edge of the convex hull.

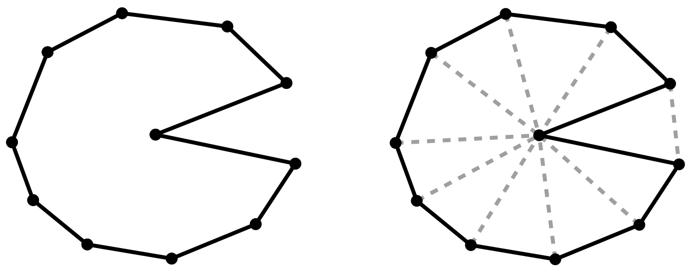

Figure 2: Left: A Hamiltonian circuit with $n-1$ vertices in convex position and one vertex lying in the interior of $\mathrm{ch}(H)$. Right: The only possible embedding preserving augmentation to a 3-connected (resp., 3-edge-connected) PSLG.

The same construction provides lower bounds for vertex-connectivity augmentation: There are 3-augmentable and 2-connected PSLGs that cannot be augmented to a 3-connected PSLG with fewer than $n-2$ new edges.

4 Augmenting the edge-connectivity from two to three

Given a graph $G = (V, E)$, two vertices $u, v \in V$ are *k-edge-connected* if they are connected by at least k edge-disjoint paths. This defines a binary relation on V, which is an equivalence relation [38]. The equivalence classes are the *k-edge-connected components* of G. By Menger's theorem, G is k-edge-connected if and only if there are k edge-disjoint paths between any two vertices, that is, if V is a single k-edge-connected component. The vertices in a 1-edge-connected component of G always induce a maximal connected subgraph of G, which is called a *component* of G, for short. Note that the vertices in a k-edge-connected component do not necessarily induce a connected subgraph of G if $k \geq 3$. In $K_{3,2}$, for instance, there are three edge-disjoint paths between the two vertices of degree 3, these two vertices form a 3-edge-connected component, even though they are not adjacent. For a graph G, let $\lambda(G)$ denote the number of 3-edge-connected components.

Lemma 4.1 *Every 2-edge-connected 3-edge-augmentable* PSLG *G admits an embedding preserving augmentation to a 3-edge-connected* PSLG *with at most $\lambda(G) - 1$ new edges.*

Proof. Since $G = (V, E)$ is 3-edge-augmentable, it can be augmented to a 3-edge-connected PSLG $G' = (V, E')$. Augment G incrementally as follows. For every edge $e \in E' \setminus E$, increment G with e if it connects two distinct 3-edge-connected components of the current PSLG. Since G is 2-edge-connected, a new edge between any two 3-edge-connected components merges those components into one. That is, every new edge decreases the number of 3-edge-connected components by at least one, so at most $\lambda(G) - 1$ new edges are added. □

The simple tool introduced in Lemma 4.1 leads to a tight bound for the number of new edges that can augment a *2-edge-connected* PSLG to become 3-edge-connected.

Theorem 1 *Every 2-edge-connected 3-edge-augmentable* PSLG *with $n \geq 4$ vertices admits an embedding preserving augmentation to a 3-edge-connected* PSLG *by adding at most $n - 2$ new edges. This bound cannot be improved.*

Proof. Let G be a 2-edge-connected 3-edge-augmentable PSLG with $n \geq 4$ vertices. If $\lambda(G) \leq n - 1$, then Lemma 4.1 completes the proof.

Assume that $\lambda(G) = n$. This means that every 3-edge-connected component consists of a single vertex. Since G is already 2-edge-connected, every edge is part of some circuit. Any two circuits either are disjoint or share exactly one vertex, since no two vertices are connected by three edge-disjoint paths. It follows that G is the union of edge-disjoint circuits, and so G is an Eulerian tour. We distinguish two cases.

Case 1. ***G is Hamiltonian.*** (Refer to Fig. 3, left) Since G is 3-edge-augmentable, its vertices are not in weakly convex position, and so there is an edge e of the convex hull $\mathrm{ch}(G)$ which is not an edge of G. The endpoints of e partition the Hamiltonian circuit G into two paths P_1 and P_2. Since both endpoints of e lie on $\mathrm{ch}(G)$, all internal vertices of one of the paths, say P_1, lie in the interior of $\mathrm{ch}(G)$. Note that G is a simple polygon, in which e is an external diagonal. In an arbitrary triangulation of the interior of this polygon, there is an edge f that connects two internal vertices of paths P_1 and P_2. Let $G' = G + \{e, f\}$. The four distinct endpoints of e and f are in a single 3-edge-connected component of G'. By Lemma 4.1, G' can be further augmented to a 3-edge-connected PSLG by adding at most $n - 4$ new edges. Together with e and f, G has an embedding preserving augmentation to a 3-edge-connected PSLG with at most $n - 2$ new edges.

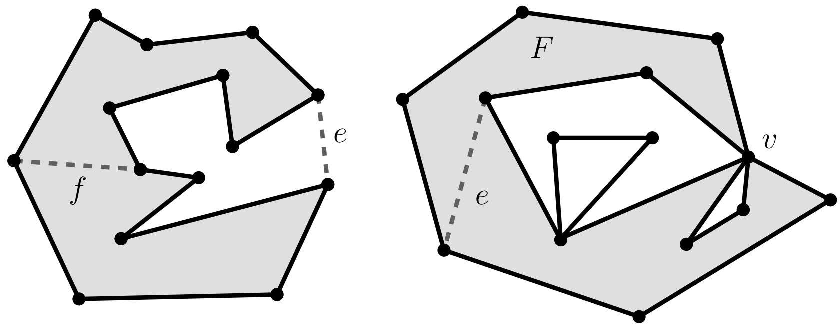

Figure 3: Left: A Hamiltonian circuit, a new edge e along the convex hull, and another edge f. Right: A non-Hamiltonian 2-edge-connected PSLG, and a face F whose boundary is not a simple polygon.

Case 2. ***G is not Hamiltonian.*** (Refer to Fig. 3, right.) Then G has a face F whose boundary is not a simple polygon, that is, some vertex v appears twice along the counterclockwise traversal of the edges incident to F. Vertex v partitions the boundary of F into at least two paths. If F is a bounded face, then in any triangulation of face F, there is an edge e that connects two internal vertices of two distinct paths. If F is the outer face, then there is an edge e that connects two internal vertices of two distinct paths. In both cases, let $G' = G + \{e\}$. Vertex v together with the two endpoints of e form a single 3-edge-connected component of G'. By Lemma 4.1, G' can be further augmented to a 3-edge-connected PSLG by adding at most $n-3$ new edges. Together with e, G has an embedding preserving augmentation to a 3-edge-connected PSLG with at most $n-2$ new edges.

The upper bound $n-2$ is best possible for the lower bound construction given in Fig. 2. □

5 Preliminaries

In the next section (Section 6), we present an algorithm for augmenting a 3-edge-augmentable PSLG with n vertices to a 3-edge-connected PSLG with at most $2n-2$ new edges. In this section, we prove a couple of auxiliary results and introduce notation for the number of bridges, components, reflex vertices, and singletons. They will play a key role in tracking the number of new edges. We also present a few simple inequalities used for verifying that at most $2n-2$ edges have been added.

Multi-edges. In the next section we present an algorithm that augments a 3-edge-augmentable PSLG to a *simple* 3-edge-connected PSLG. In intermediate steps of our augmentation algorithm, we may use a *planar straight line multi-graph* (PSLMG, for short), where the multiplicity of each edge is a positive integer. In the course of the augmentation algorithm, some new edges may be parallel to original or previously added edges of the graph. Since every edge is embedded as a straight line segment, parallel edges are represented by the same line segment. For a PSLMG G, we denote by $\widehat{G}$ the PSLG obtained by changing the multiplicity of every edge of G to one. The following proposition allows replacing multi-edges by single edges without decreasing the edge-connectivity. Abellanas *et al.* [1, Lemma 4] proved an analogous result for 2-edge-connected PSLMGs.

Proposition 5.1 *Let G be a 3-edge-connected* PSLMG *such that $\widehat{G}$ is 3-edge-augmentable, and let e be an edge of multiplicity at least 2 in G. Then we can obtain a 3-edge-connected* PSLMG *from G by decrementing the multiplicity of e by one and adding at most one new edge of multiplicity 1.*

Proof. Decrement the multiplicity of e by one, and denote the resulting PSLMG by G'. If G' is 3-edge-connected, then our proof is complete. Suppose that G' is not 3-edge-connected. Then G' has a 2-bridge (possibly an edge of multiplicity 2). Every 2-bridge in G' must contain e, and an edge can be part of at most one 2-bridge of G'. It follows that G' has a unique 2-bridge, which contains e. The deletion of this 2-bridge decomposes G' into two subgraphs, say G'_1 and G'_2. Since $\widehat{G}$ is 3-edge-augmentable, it can be augmented to a 3-edge-connected PSLG G''. At least three edges of G'' connect $V(G_1)$ and $V(G_2)$, one of which is not present in $\widehat{G}$. Denote this edge by f. Now the PSLMG $G' \cup \{f\}$ has no 2-bridge, hence it is 3-edge-connected. □

Corollary 5.2 *If a* PSLG *G has an embedding preserving augmentation to a 3-edge-connected* PSLMG *with m new edges (possibly duplicates), then G has an embedding preserving augmentation to a* PSLG *with at most m new (distinct) edges.*

Proof. Augment G to PSLMG G', increasing the total multiplicity by m. Apply Proposition 5.1 successively while G' has an edge of multiplicity more than one. We obtain a 3-edge-connected PSLG with all the original edges of G and at most m additional edges. □

We also extend Lemma 4.1 to PSLMGs.

Lemma 5.3 *Let G be a 2-edge-connected* PSLMG *such that $\widehat{G}$ is 3-edge-augmentable. Then G admits an embedding preserving augmentation to a 3-edge-connected* PSLMG *with at most $\lambda(G) - 1$ new edges.*

Proof. Since $\widehat{G} = (V, E)$ is 3-edge-augmentable, it can be augmented to a 3-edge-connected PSLG $G' = (V, E')$. Augment G incrementally as follows. For every edge $e \in E' \setminus E$, increment G with e if it connects two distinct 3-edge-connected components of the current PSLMG. Since G is 2-edge-connected, a new edge between any two 3-edge-connected components merges those components into one. That is, every new edge decreases the number of of 3-edge-connected components by at least one, so at most $\lambda(G) - 1$ new edges are added. □

Notation for bridges, components, and vertices along the convex hull. Consider a PSLG $G = (V, E)$. A each vertex $v \in V$, the *rotation order* of the incident edges is the cyclic order in which the edges appear counterclockwise around v. If two edges $e_1, e_2 \in E$ are adjacent at vertex $v \in V$, and they are consecutive in the rotation order of V, then they are both adjacent to a common face of G, and determine an *angle* $\angle(e_1, e_2)$ of G with apex v. A vertex $v \in V$ is *reflex* if it is the apex of an angle of at least 180°. For example, a vertex of degree 1 or 2 is always reflex, but a singleton is not reflex. If a vertex of degree 2 is incident to two collinear edges, we designate one adjacent angle of 180° as reflex, and the other one as convex, arbitrarily. With this convention, every reflex vertex is the apex of a reflex angle in a unique face.

An edge in a graph G is a *bridge* if the deletion of the edge disconnects one of the connected components of G. Similarly a pair of edges in a graph G is a *2-bridge* (or 2-edge-cut) if the deletion of both edges disconnects one of the connected components of G.

By Euler's formula, a PSLG with n vertices has at most $2n - 5$ bounded faces. We use a stronger inequality that includes the number of reflex vertices.

Lemma 5.4 ([36]) *Let G be a* PSLG *with f bounded faces and n vertices, r of which are reflex. Then we have $f + r \leq 2n - 2$.*

This bound cannot be improved: it is tight, for example, for triangulations. Applying Lemma 5.4 for each 2-edge-connected component of a PSLG G, independently, we can conclude the following.

Corollary 5.5 *Let G be a* PSLG *with b bridges, c non-singleton components, f bounded faces, n vertices, r of which are reflex, and s singletons. Then*

$$b + c + f + r + 2s \leq 2n, \tag{1}$$

with equality if and only if G is a forest having only reflex vertices.

Proof. Let G_0 be the graph obtained from G by removing all bridges. Let $b_0 = 0$, c_0, f_0, r_0, and s_0 denote the corresponding parameters of G_0 (e.g., c_0 denotes the number of non-singleton connected components of G_0). Applying Lemma 5.4 for each non-singleton component of G_0, we have $f_0 + r_0 \leq 2(n - s_0) - 2c_0$. After adding $2s_0$ to both sides, we have $f_0 + r_0 + 2s_0 \leq 2n - 2c_0$, thus $b_0 + c_0 + f_0 + r_0 + 2s_0 \leq 2n - c_0$.

We now successively add the bridges of G, in an arbitrary order, to G_0. Whenever we add a bridge, the number of components (including both singleton and non-singleton components) decreases by one, the number of bridges increases by one, and the number of faces remains unchanged. If an endpoint of the bridge is a singleton, then the singleton becomes a reflex vertex. Hence after adding a bridge, the quantity $b + c + f + r + 2s$ remains the same or decreases (some reflex points may become non-reflex). Therefore, $b + c + f + r + 2s \leq 2n$, with equality if and only if $c_0 = 0$ (that is G is a forest) and all vertices are reflex. □

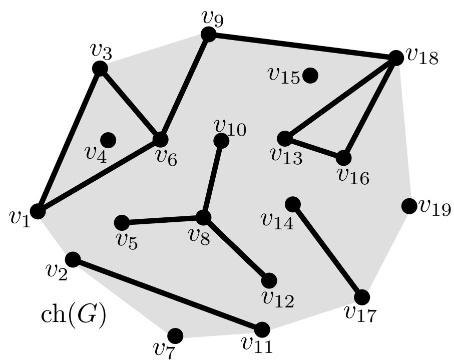

Figure 4: A PSLG G with 19 vertices and 13 edges. We have $b_h = 1$ (bridge v_9v_{18} is on the convex hull), $g_h = 2$ (edges v_1v_3 and v_9v_{18}), $r_h = 7$ (reflex vertices v_1, v_2, v_3, v_9, v_{11}, v_{17} and v_{18}), $s_h = 2$ (singletons v_7 and v_{19}), and $c_h = 3$ (the three components that contain v_1, v_2 and v_{17}, respectively).

After each step of our augmentation algorithm (in Section 6 below), we will derive an upper bound for the number of newly added edges in terms of b, c, f, r, and s. Inequality (1) will ensure that altogether at most $2n - 2$ edges are added.

We distinguish two types of bridges, edges, reflex vertices, singletons, and non-singleton components. (See Fig. 4.) In the input graph G, let b_h (resp., g_h, r_h, and s_h) denote the number of bridges (resp., edges, reflex vertices, and singletons) along the convex hull $\mathrm{ch}(G)$. Clearly, we have $b_h \leq g_h$. Let c_h be the number of non-singleton components with at least one vertex lying on the convex hull. Let $b_i = b - b_h$, $c_i = c - c_h$, $r_i = r - r_h$, and $s_i = s - s_h$.

Connecting singletons. To raise the edge-connectivity of a PSLG G to 3, the degree of every singleton has to increase to at least 3. We can charge at most two new edges to each singleton (i.e., the term $2s$ in Inequality (1)). We will charge the additional edges at singletons to faces and reflex vertices (i.e., the terms $f + r$ in Inequality (1)). Every vertex of degree 2 is reflex, and we will charge one new edge per reflex vertex to the term r in Inequality (1). The greatest challenge in designing our augmentation algorithm is to add r

new edges at reflex vertices that serve two purposes: they (i) connect each reflex vertex to another vertex and (ii) connect a possible nearby singleton to the rest of the graph.

Our augmentation algorithm will work in 7 stages. By the end of stage 4 of our algorithm, we obtain a PSLMG G_4 that consists of a 2-edge-connected component and some singletons lying in the interior of $\mathrm{ch}(G)$. Some of the edges of G_4, which have been added in stages 1-4, are labeled *deformable*. These are not part of the input graph, and so they can be changed. We will compute the following structure in stage 5.

> (♡) The interior of $\mathrm{ch}(G_5)$ is decomposed into pairwise disjoint convex *regions*, $C_1, C_2, \ldots, C_\ell$. For every $j = 1, 2, \ldots, \ell$, there is a deformable edge e_j whose endpoints lie on the boundary of C_j; the deformable edges e_j are distinct; and the only edge of G_5 that may intersect the interior of C_j is e_j.

In every convex region C_j, $j = 1, 2, \ldots, \ell$, we can replace the deformable edge $e_j = u_j v_j$ by a path between u_j and v_j that lies entirely in C_j and passes through all singletons in C_j. The resulting PSLG is 2-edge-connected, and so we can apply Lemma 5.3 to obtain a 3-edge-connected PSLG.

Deformable edges. In stages 1-4 of the algorithm, we maintain a unique deformable edge $\tau(F)$ for each bounded face F of the current PSLMG. The edge $\tau(F)$ is a new edge added during augmentation and it may be parallel to an edge of the input PSLG G. In stage 5, we decompose the interior of $\mathrm{ch}(G)$ into convex regions with property (♡), with one deformable edge assigned to each convex region, and in stage 6, each deformable edge $u_j v_j$ will be replaced by a path between u_j and v_j.

In stage 1, we add a deformable edge parallel to an existing edge for every bounded face of G. In stages 2-5, we maintain the deformable edges $\tau(F)$ according to the following rule:

> Rule ($*$): If a new edge e decomposes a face F into two faces, F_1 and F_2, we define $\tau(F_1)$ and $\tau(F_2)$ as follows. If F is a bounded face and, w.l.o.g., $\tau(F)$ is adjacent to F_1, then let $\tau(F_1) = \tau(F)$ and $\tau(F_2) = e$. If F is the outer face and, without loss of generality, F_1 is a bounded face (hence F_2 is the new outer face), then let $\tau(F_1) = e$.

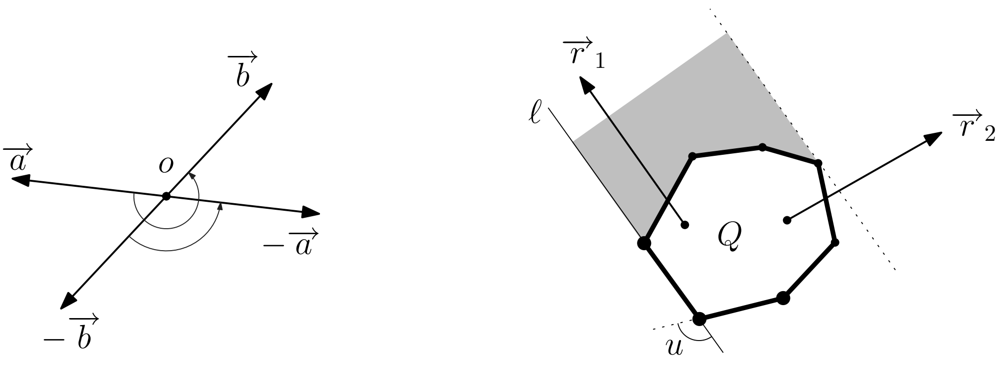

Figure 5: Left: A reflex angular domain $\angle(\vec{a}, \vec{b})$ and its reverse wedge $\angle(-\vec{b}, -\vec{a})$. Right: A convex polygon Q and two rays. The reverse wedges of the exterior angles at u_1, u_2, and u_3 are disjoint from both rays.

Reverse wedges. We conclude this section with a technical lemma (Lemma 5.6). A *wedge* (or angular domain) $\angle(\vec{a}, \vec{b})$, defined by rays $\vec{a}$ and $\vec{b}$ emanating from a point o, is the region swept by the ray rotated about o counterclockwise from position $\vec{a}$ to $\vec{b}$. Refer to Fig. 5. If the angular domain $\angle(\vec{a}, \vec{b})$ is reflex, we define its *reverse wedge* to be $\angle(-\vec{b}, -\vec{a})$.

Lemma 5.6 *Let Q be a convex polygon, and let $\vec{r}_1$ and $\vec{r}_2$ be two rays emanating from Q. Then Q has a vertex u such that the reverse wedge of the exterior angle of Q at u is disjoint from the relative interiors of $\vec{r}_1$ and $\vec{r}_2$.*

Proof. Consider the two lines tangent to Q and are parallel to $\vec{r}_1$. At least one of them, say line ℓ, is disjoint from the relative interior of $\vec{r}_2$. If exactly one vertex of Q lies on ℓ, then let u be this vertex. Otherwise let u be the first vertex of Q along the line ℓ, oriented in the same way as $\vec{r}_1$. □

6 Augmentation algorithm

Let $G = (V, E)$ be a 3-edge-augmentable PSLG with $n \geq 4$ vertices. We augment G to a 3-edge-connected PSLG with at most $2n - 2$ new edges. The seven main stages of our augmentation algorithm are outlined below. At the end of stage $j = 1, 2, \ldots, 7$, the input G has been augmented to a PSLMG G_j, where G_7 is a 3-edge-connected PSLG. For stage $j = 2, 3, \ldots, 7$, we note the key properties of the resulting PSLMG G_j in brackets.

1. Create a deformable edge for each bounded face.
2. Add all convex hull edges [all hull edges are in a single component].
3. Connect all non-singleton components [one big component and singletons].
4. Eliminate bridges [a big 2-edge-connected component and singletons].
5. Add a new edge at each reflex vertex in the interior of $\mathrm{ch}(G)$ [singletons and a big 2-edge-connected component, where every 3-edge-connected component is incident on the outer face].
6. Connect singletons lying in the interior of $\mathrm{ch}(G)$ [a 2-edge-connected PSLMG].
7. Apply Lemma 5.3 [a 3-edge-connected PSLG].

Stage 1. Deformable edges for all bounded faces. For every bounded face F in G, add a deformable edge $\tau(F)$ parallel to an arbitrary edge in E adjacent to F. We have created $\boxed{f}$ deformable edges, each of which is parallel to an existing edge of G. We obtain a PSLMG G_1.

Stage 2. Convex hull edges. Augment G_1 successively with the edges of the convex hull $\mathrm{ch}(G)$ if they are not already present in G_1. The number of vertices along the convex hull is $r_h + s_h$, and so we have added $\boxed{r_h + s_h - g_h}$ new edges.

For $j = 2, \ldots 7$, the boundary of the outer face in G_j will be a simple polygon, which we denote by P_j. Let $\lambda_h(G_j)$ denote the number of 3-edge-connected components of G_j that have at least one vertex incident on P_j. (Recall that $\lambda(G)$ denotes the total number of 3-edge-connected components of a graph G, *c.f.* Section 4). We will use the following lemma in stage 7.

Lemma 6.1 *At the end of stage 2, we have $\lambda_h(G_2) \leq c_h + s_h + g_h$.*

Proof. The original PSLG G has $c_h + s_h$ components with at least one vertex lying on $\text{ch}(G)$. If we remove all g_h convex hull edges from G, the remaining PSLG has at most $c_h + s_h + g_h$ components with at least one vertex on $\text{ch}(G)$. All hull vertices in each of these components are 3-edge-connected in G_2. Hence $c_h + s_h + g_h$ is an upper bound on $\lambda_h(G_2)$. □

Stage 3. Connecting non-singleton components. In the rest of the paper, let K always denote the connected component of the current PSLMG that contains the boundary of the outer face. In this stage, we incrementally add new edges between K and all other non-singleton components, which lie in the interior $\text{ch}(G)$. Repeat the following procedure, which augments G_2 with one or two new edges, until K becomes the only non-singleton component.

Let H denote the disjoint union of all non-singleton components of G_2 lying in the interior of $\text{ch}(G)$. Let U denote the set of vertices of $\text{ch}(H)$. Refer to Fig. 6. Pick an arbitrary vertex $u \in U$. It is clear that u is a reflex vertex. The ray emitted from u along the bisector of the reflex angle hits some an edge vw of K. Let $x \in uv$ be the point hit by the ray. Denote by $\text{path}(u, v)$ and $\text{path}(u, w)$, respectively, the shortest paths from u to v and to w which are homotopy equivalent to the paths (u, x, v) and (u, x, w) within a face of K (note that these paths may cross edges of H, which are not part of K).

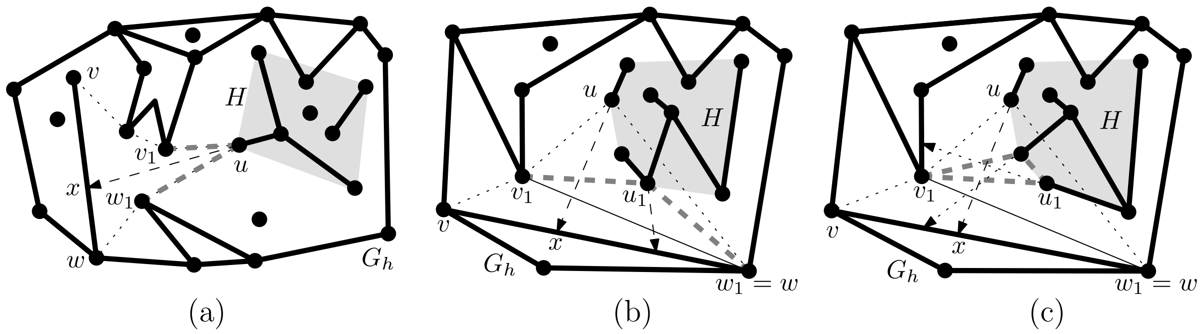

Figure 6: The shortest paths $\text{path}(u, v)$ and $\text{path}(u, w)$ and edge vw form a pseudo-triangle. (a) The interior of triangle $\Delta(u, v_1, w_1)$ is disjoint from U. (b) The bisector ray of the reflex angle at u_1 hits v_1w_1. (c) The bisector ray of the reflex angle at u_1 hits uv_1

Let v_1 and w_1 be the vertex of $\text{path}(u, v)$ and $\text{path}(u, w)$, respectively, adjacent to u (possibly, $v_1 = v$ or $w_1 = w$). If the interior of triangle $\Delta = \Delta(u, v_1, w_1)$ is disjoint from U (Fig. 6(a)), then augment G_2 with the edges uv_1 and uw_1. This decreases the number of non-singleton components by one, and it also decreases the number of reflex vertices by one because u is no longer reflex.

Suppose now that the interior of triangle Δ intersects U. Let u_1 be the vertex in $U \cap \text{int}(\Delta)$ closest to the supporting line of v_1w_1. Note that the points in $U \cap \text{int}(\Delta)$ must lie on one side of the ray $\overrightarrow{ux}$, since U is in convex position and $u \in U$. Assume w.l.o.g. that $U \cap \text{int}(\Delta)$ and w lie on the same side of $\overrightarrow{ux}$.

If the bisector ray of the reflex angle of u_1 crosses v_1w_1 (Fig. 6(b)), then we augment G_2 with the edges u_1v_1 and u_1w_1. These edges are already outside of $\text{ch}(H)$. Similarly to the previous case, the number of non-singleton components drops by one, and the number of reflex vertices also drops by one because u_1 is no longer reflex.

If, however, the bisector ray of the reflex angle of u_1 crosses uv_1 (Fig. 6(c)), then there are two consecutive vertices along $\text{ch}(H)$ between u and u_1, say u' and u'', such that the bisector ray of u' hits v_1w_1 and the bisector ray of u'' hits uv_1. In this case, we add three edges $u'v_1$, $u'u''$, and $u''v_1$. The number of non-singleton components drops by at least one, and the number of reflex vertices drops by two (specifically, u' and u'' are no longer reflex).

After iterating, the resulting PSLMG G_3 has one non-singleton component (which contains all hull edges) and all singletons lie in some bounded faces. Note also that none of the new edges are bridges. Let r_3 denote the number of reflex vertices of G_2 that are no longer reflex in G_3. We have used at most $\boxed{c_i + r_3}$ new edges. All c_i non-singleton components have been connected, and the number of reflex vertices in the interior of $\mathrm{ch}(G)$ has dropped by r_3.

Stage 4. Eliminate bridges. Replace every bridge lying inside $\mathrm{ch}(G)$ by a double edge. The total multiplicity of edges increases by at most $\boxed{b_i}$. We obtain a PSLMG G_4, which consists of one 2-edge-connected component K and s_i singletons lying in the interior of $\mathrm{ch}(G)$.

Stage 5. Adding new edges at reflex vertices in the interior of $\mathrm{ch}(G)$. Arguably, this is the most complicated stage of our augmentation algorithm. At the beginning of this stage, we have a PSLMG G_4 that consists of a 2-edge-connected PSLMG K (which contains all hull edges) and some singletons. There are at most $r_i - r_3$ reflex vertices in the interior of $\mathrm{ch}(G)$. We modify K (by adding some new edges and "deforming" some of the deformable edges) to a PSLMG where every 3-edge-connected component is incident to the outer face, and increase the number of edges by at most $\boxed{r_i - r_3}$. We also compute a decomposition of the interior of $\mathrm{ch}(G)$ into convex regions with property (♡).

Let R initially be the set of reflex vertices in the interior of $\mathrm{ch}(G)$. We have $|R| \leq r_i - r_3$. Algorithm 1 (below) will *process* the vertices in R, and increase the number of edges by one for each $u \in R$. It either adds a new edge uv, or replaces some deformable edge vw by two new edges vu and uw. Every vertex of R is processed just once, and then it is immediately removed from R, even if it remains a reflex vertex. Algorithm 1 will maintain a set of bounded faces $\mathcal{F}$, and a deformable edge $\tau(F)$ for every face $F \in \mathcal{F}$. Initially, $\mathcal{F}$ contains *all* bounded faces of G_4. As the algorithm proceeds, however, F may cease to contain all of the bounded faces. In addition, $\mathcal{F}$ may possibly contain a 2-gon (in case a new edge is parallel to some edge of G_4).

We introduce some notation. Let $u \in V$ be a reflex vertex of G_4 lying in the interior of $\mathrm{ch}(G)$. Let W_u denote the reverse wedge of the reflex angle at u in G_4, let $\vec{a}_u$ be the bisector ray of the reflex angle at u in G_4. Let F_u denote the face at u containing some initial portion of the ray $\vec{a}_u$, and let x_u be the point where $\vec{a}_u$ hits the boundary of F_u. Wedge W_u and ray $\vec{a}_u$ are defined for the input PSLMG of Algorithm 1, and do not change during the algorithm, however, face F_u and x_u may change when the algorithm adds or deforms edges. We define orientation for every reflex vertex u with respect to the deformable edge $\tau(F_u)$. We say that a reflex vertex u is *ccw* (resp., *cw*) if $x_u \notin \tau(F_u)$ and the triple $(\tau(F_u), u, x_u)$ is in counterclockwise (resp., clockwise) order along the boundary of F_u. Visibility is defined with respect to the current (augmented) PSLMG, where all edges are opaque: a point p is *visible* to point q if the relative interior of segment pq is disjoint from the edges of the PSLMG.

Algorithm 1

Input: A PSLMG $G_4 = (V, E_4)$ that consists of some singletons and a 2-edge-connected component such that the boundary of the outer face is a simple polygon $P_4 = \mathrm{ch}(V)$; and a function τ that maps a unique edge $\tau(F)$ to every bounded face F of G_4.

Output: $G_5 = (V, E_5)$.

Set $R :=$ the set of reflex vertices lying in the interior of P_4. Compute W_u and $\vec{a}_u$ with respect to G_4 for every vertex $u \in R$ (W_u and $\vec{a}_u$ are fixed during the whole algorithm). Let $\mathcal{F}$ be the set of all bounded faces of G_4. Set $E_5 := E_4$.

while $R \neq \emptyset$ **do**

if there is a vertex $u \in R$ that sees a non-singleton vertex v in W_u (Fig. 7(a-b)), **then**

set $E_5 := E_5 + \{uv\}$ and $R := R \setminus \{u\}$. Edge uv splits face $F_u \in \mathcal{F}$ into two faces. Update $\mathcal{F}$ by replacing F_u with the new faces. Following Rule ($*$), edges $\tau(F_u)$ and uv become the deformable edges of the two new faces. If $v \in R$ and uv splits the reflex angle at v into two convex angles, then set $R := R \setminus \{v\}$.

else if there is a vertex $u \in R$ such that $\vec{a}_u$ does not hit $\tau(F_u)$ (Fig. 7(c-d)), **then**

among all vertices $u \in R$ where $\vec{a}_u$ does not hit $\tau(F_u)$, pick a vertex $u \in R$ which is either the first ccw reflex vertex along the boundary of F_u starting from $\tau(F_u)$ in ccw direction or the first cw reflex vertex of F_u starting from $\tau(F_u)$ in cw direction. Let vw be the edge hit by $\vec{a}_u$ such that v is on the same side of $\vec{a}_u$ as $\tau(F_u)$. Compute the shortest path $\mathrm{path}(u, w)$ in F_u, and denote its vertices by $\mathrm{path}(u, w) = (u = p_0, p_1, p_2, \ldots, p_\ell = w)$. Set $E_5 := E_5 + \{p_j p_{j+1} : 0 \leq j \leq \ell - 1\}$ (possibly increasing the multiplicity of some edges) and $R := R \setminus \{p_j : 0 \leq j \leq \ell - 1\}$. If $w \in R$ and $p_{\ell-1}w$ splits the reflex angle at w into two convex angles, then set $R := R \setminus \{w\}$. The new edges split $F_u \in \mathcal{F}$ into $\ell + 1$ new faces (some of which may be a 2-gon). Update $\mathcal{F}$ by replacing F_u with the new faces. Following Rule ($*$), edges $\tau(F_u)$ and $p_j p_{j+1}$, $0 \leq j \leq \ell - 1$, become the deformable edges of the new faces in $\mathcal{F}$.

else

for every $u \in R$, ray $\vec{a}_u$ hits edge $\tau(F_u)$ (Fig. 7(d-e)). Pick a vertex $u \in R$ such that the distance between u and the supporting line of edge $\tau(F_u)$ is minimal. Let $vw = \tau(F_u)$. Set $E_5 := (E_5 \setminus \{\tau(F_u)\}) + \{uv, uw\}$ and $R := R \setminus \{u\}$. The new edges split $F_u \in \mathcal{F}$ into 3 new faces. Update $\mathcal{F}$ by replacing F_u with the two new faces adjacent to uv or uw (but not both), and let their deformable edges be uv and uw, respectively. If $\tau(F_u)$ is parallel to an edge of the original graph G, then the triangular face (u, v, w) is a new face, which is not in $\mathcal{F}$. Otherwise, the removal of edge $\tau(F_u)$ merges triangle (u, v, w) with a face F' lying on the opposite side of vw; update $\mathcal{F}$ by replacing F' with this new face.

end if

end while

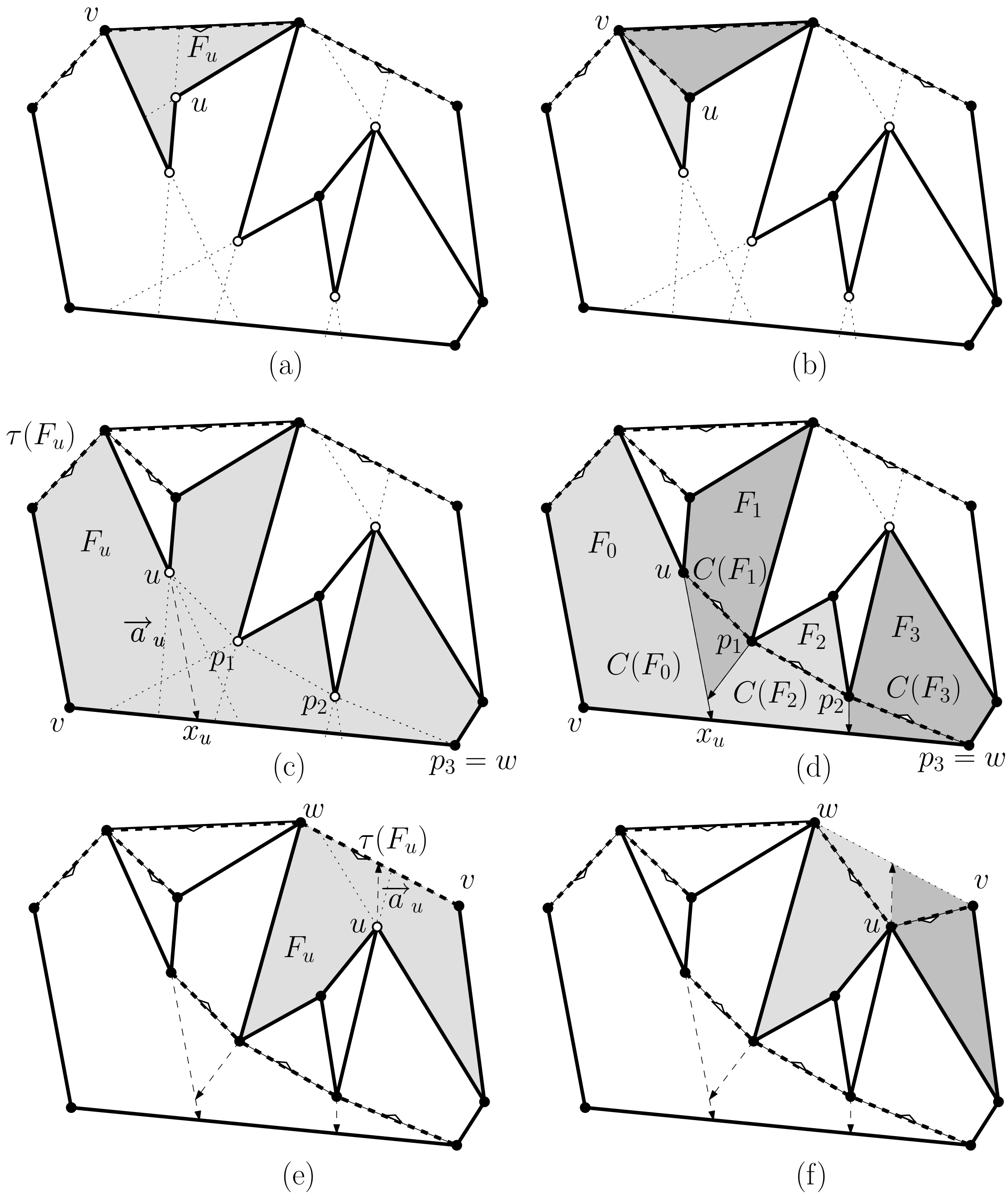

Figure 7: The three rows show three consecutive *while* loops of Algorithm 1. Deformable edges are marked with dashed lines. Reflex vertices in R are marked with little circles. (a) A PSLG G_4, and a vertex $u \in R$ that sees a non-singleton vertex v in W_u. (b) We augment the graph with uv. (c) Vertex $u \in R$ where $\vec{a}_u$ does not hit $\tau(F_u)$. (d) We augment the graph with all edges along the chain (u, p_1, p_2, p_3). (e) Vertex $u \in R$ where $\vec{a}_u$ hits $\tau(F_u)$. (f) We deform $\tau(F_u) = vw$ into the path (v, u, w).

We refer to the three cases in the while loop of Algorithm 1 as *case 1*, *case 2*, and *case 3*, respectively.

Lemma 6.2 *The execution of the while loop in Algorithm 1 is partitioned into three consecutive* phases *(some of which may be empty) such that case i applies to all vertices $u \in R$ processed in phase i , for $i = 1, 2, 3$.*

Proof. Case 1 applies for a vertex $u \in R$ that sees a non-singleton vertex in W_u. Since no edges are removed in Cases 1 and 2, no new visibilities occur during applications of these cases. It follows that, in general, the while loop in Algorithm 1 starts with zero or more applications of Case 1, followed by zero or more applications of Case 2, and then (possibly) by the first application of Case 3. If Case 3 is not applied during Algorithm 1, then the lemma clearly holds.

Just before the first application of Case 3, we have the following two properties: (1) For every $u \in R$, the ray $\vec{a}_u$ hits edge $\tau(F_u)$; otherwise Case 2 would apply for u. (2) Every ray emanating from u within wedge W_u also hits edge $\tau(F_u)$; otherwise Case 1 would apply for u. If Case 3 applies for some $u \in R$, then for every $v \in R$, $v \neq u$, the rays emanating from v within W_v will continue to hit $\tau(F_v)$, even if F_v becomes smaller and the original $\tau(F_v)$ is replaced by two edges (this happens if $F_u = F_v$ before applying Case 3 for u). Thus, properties (1) and (2) are maintained in the remainder of Algorithm 1, and so only Case 3 may be applied. The lemma follows. □

Lemma 6.2 means that when we consider Case 1, we may assume that Cases 2 and 3 have not occurred before; and when we consider Case 2, we may assume that Case 3 has not occurred before.

The following lemma, describes the non-singleton components of the output of Algorithm 1. Note that Algorithm 1 does not add edges incident to any singletons of G_4, and so the G_4 and G_5 have the same singletons.

Lemma 6.3 *Algorithm 1 outputs a* PSLG *G_5 such that, apart from possible singletons, the boundary of the outer face is a simple polygon P_5, and every 3-edge-connected component is incident on P_5.*

Proof. Consider the output $G_5 = (V, E_5)$ of Algorithm 1. Note that Algorithm 1 does not necessarily *augment* G_4 to G_5, since it may replace a deformable edge $vw = \tau(F_u)$ by a path (v, u, w). In the course of several steps, an edge $vw = \tau(F_u)$ of G_4 may "evolve" into a simple path between v and w. In particular, for any partition of the vertex set V, the number of edges among the subsets of vertices cannot decrease.

We have $P_4 = \mathrm{ch}(G)$. The boundary of the outer face is modified during the algorithm if a ray $\vec{a}_u$ of a vertex $u \in R$ hits an edge $\tau(F_u) = vw$ along the outer face, and the edge vw is replaced by the edges uv and uw. Then vertex u becomes a vertex of the outer face and is removed from R. Every such step maintains the property that the boundary of the outer face is a simple polygon, and no point of R lies on this boundary. Moreover, $\mathrm{int}(P_5) \subseteq \mathrm{int}(P_4)$.

Next we show that every 3-edge-connected component of G_5 is incident on the outer face. Suppose, to the contrary, that there is a 2-bridge $\{e, f\}$ in G_5 such that both e and f are inside P_5. The graph $G_5 - \{e, f\}$ has two connected components, which we denote by $G_5(U)$ and $G_5(V \setminus U)$ with vertex sets U and $V \setminus U$, respectively. We may assume without loss of generality that U lies in the interior of P_5 (and $V \setminus U$ contains all vertices of P_5). Hence U also lies in the interior of P_4. Since G_4 is 2-edge-connected, there are exactly two edges, say e_0 and f_0, between U and $V \setminus U$ in G_4 (Fig. 8, left). We may assume that either $e_0 = e$ or e_0 evolves to a path that contains e; and similarly, either $f_0 = f$ or f_0 evolves to a path that contains f.

Since G_4 is 2-edge-connected, the minimum vertex degree in both G_4 and G_5 is at least 2. Every vertex of degree 2 is reflex. Algorithm 1 increased the degree of every vertex of degree 2 lying in the interior of P_4 to at least 3. It follows that $G_5(U)$ cannot be a singleton (which would be incident to e and f only), or

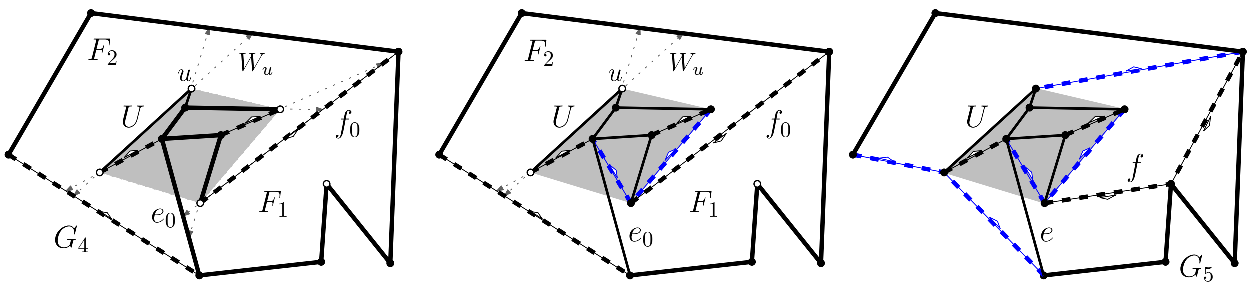

Figure 8: Left: A 2-bridge $\{e_0, f_0\}$ in G_4. Middle: Algorithm 1 has processed two vertices and $\{e_0, f_0\}$ remains a 2-bridge. Right: Algorithm 1 terminates, and the edges e_0 and f_0 evolve into paths containing the edges $\{e, f\}$ between U and $V \setminus U$.

a path of collinear edges (where each endpoint would be incident to the path and either e or f). Therefore, $G_5(U)$ has at least three non-collinear vertices, and so $\mathrm{ch}(U)$ has at least three extremal vertices.

The PSLG G_4 has exactly two faces adjacent to both U and $V \setminus U$, say faces F_1 and F_2, each of which is adjacent to both e_0 and f_0. Every vertex on the convex hull $\mathrm{ch}(U)$ is incident to F_1 or F_2 (or both). Algorithm 1 modifies edges e_0 or f_0 only in phase 3, if they are the special edges $\tau(F_1)$ or $\tau(F_2)$, and all rays $\vec{a}_u$, $u \in R$, lying in F_1 and F_2 hit these edges.

By Lemma 5.6, $\mathrm{ch}(U)$ has an extremal vertex u such that the reverse wedge of the exterior angle of $\mathrm{ch}(U)$ at u does not intersect e_0 or f_0 (except possibly in u). Vertex u is reflex in G_4, it lies in the interior of P_4, and so it is initially in R. Since the edges of G_4 incident on u lie on or inside $\mathrm{ch}(U)$, the wedge W_u is part of the reverse wedge of $\mathrm{ch}(U)$ at u. Hence $\vec{a}_u$ hits some edge spanned by $V \setminus U$.

We distinguish three cases depending on the phase in which Algorithm 1 processes u. If u is processed in phase 1, then it is connected to a vertex in $V \setminus U$. If u is processed in phase 2, then $\vec{a}_u$ hits some edge vw with $v, w \in V \setminus U$, and we add all edges along the a geodesic $\mathrm{path}(u, w)$, including a new edge between U and $V \setminus U$. If u is processed in phase 3, ray $\vec{a}_u$ hits the edge $\tau(F_u)$ between two vertices of $V \setminus U$, which is necessarily different form e_0 and f_0, and $\tau(F_u)$ evolves into a path through u. In all three cases, we add a new edge between U and $V \setminus U$. This contradicts the assumption that $\{e, f\}$ is a 2-bridge in G_5. □

Lemma 6.4 *At the end of stage 5, we have* $\lambda(G_5) \leq c_h + g_h + s$.

Proof. At the end of stage 2, we have $\lambda_h(G_2) \leq c_h + g_h + s_h$ by Lemma 6.1. All convex hull edges remain part of our graph in stages 3-4, and the addition of new edges can only merge some of the 3-edge-connected components incident on the convex hull. So at the end of stage 4 we still have $\lambda_h(G_4) \leq c_h + g_h + s_h$. Since every 3-edge-connected component of G_5 is either one of the s_i singletons or incident on the outer face P_5, we have $\lambda(G_5) = \lambda_h(G_5) + s_i$. It is enough to show that $\lambda_h(G_5) \leq \lambda_h(G_4)$.

Assume that P_4 is the boundary of the outer face in G_4. Algorithm 1 may change P_4 by replacing an edge vw of P_4 with the edges vu and uw for some reflex vertex $u \in \mathrm{int}(P_4)$. In each such step, a new vertex u appears along the outer face. By Lemma 6.3, the vertex u is connected to another vertex of the outer face by a path that lies in the interior of P_5. So this deformation does not increase the number of 3-edge-connected components incident on the outer face, and so we have $\lambda_h(G_5) \leq \lambda_h(G_4)$. □

Lemma 6.5 *Algorithm 1 increases the number of edges by at most* $r_i - r_3$.

Proof. It is enough to show that in each step, we add at most one new edge for each vertex removed from R, since $|R| \leq r_i - r_3$. This is clearly true for Case 1 and Case 3. In Case 2, we add ℓ new edges, so we need to show that ℓ vertices were removed from R. Vertex $u = p_0$ was removed from R. It is enough to check that all reflex vertices $p_1, p_2, \ldots, p_{\ell-1}$ are in R at the beginning of the step.

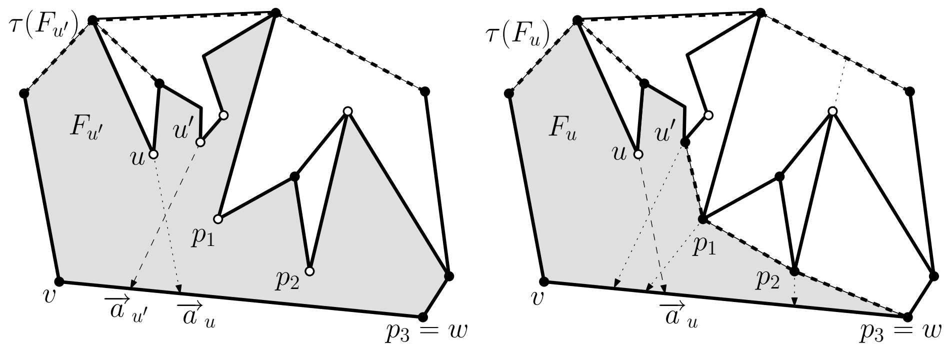

Figure 9: We suppose, by contradiction, that Case 2 is first applied for a vertex $u' \in R$ (left); and then Case 2 is applied for vertex $u \in R$ (right) but the *reflex vertices* p_1 and p_2 are no longer in R.

Suppose that some reflex vertex p_i, $1 \le i \le \ell - 1$, is *not* in R. Recall that in phase 2, only Cases 1 or 2 has occurred before (Lemma 6.2), so no deformable edge has been deformed yet. The reflex vertices processed in phase 1 are no longer reflex in phase 2. That is, p_i has been processed in an earlier step of phase 2. There was a reflex vertex u' (possibly $u' = p_i$) that did not see any vertex in its wedge $W_{u'}$, ray $\vec{a}_{u'}$ did not hit $\tau(F_{u'})$, and the path $\mathrm{path}(u', w)$ passed through vertex p_i. Refer to Fig. 9. However, then $\vec{a}_{u'}$ and $\vec{a}_u$ hit the same edge vw. Ray $\vec{a}_{u'}$ splits $F_{u'}$ into two parts, where u is incident to the part adjacent to $\tau(F_{u'})$. This contradicts the choice of u' (that is, $u \in R$ would have been chosen instead of $u' \in R$). This proves that vertices p_i, $i = 1, 2, \ldots, \ell - 1$, are in R at the beginning of the step. □

Construction of a convex cells with property (♡). We construct a tiling of $\mathrm{ch}(G)$ with property (♡) simultaneously with Algorithm 1. We maintain a (not necessarily convex) region $C(F)$ for every face $F \in \mathcal{F}$. Initially, $C(F)$ is the interior of face F for every $F \in \mathcal{F}$. Let $\mathcal{C} = \{C(F) : F \in \mathcal{F}\}$. During Algorithm 1, the following invariants are maintained:

($\mathbf{I}_1$) The regions in $\mathcal{C}$ are interior disjoint, and they tile $\mathrm{ch}(G)$;
($\mathbf{I}_2$) for every $F \in \mathcal{F}$, the set of reflex angles of $C(F)$ equals the set of reflex angles of F with apex in R;
($\mathbf{I}_3$) for every $F \in \mathcal{F}$, the endpoints of the deformable edge $\tau(F)$ lie on the boundary of $C(F)$;
($\mathbf{I}_4$) for every $F \in \mathcal{F}$, the only edge that possibly intersects the interior of $C(F)$ is $\tau(F)$.

An immediate corollary of the invariant $\mathbf{I}_1$ is that at the end of Algorithm 1 (when $R = \emptyset$), $\mathcal{C}$ is a *convex* subdivision of $\mathrm{ch}(G)$.

Lemma 6.6 *We can maintain a set $\mathcal{C}$ of regions with invariants $\mathbf{I}_1$–$\mathbf{I}_4$ through Algorithm 1.*

Proof. Initially, for every bounded face F in G_4, we set $C(F) := \mathrm{int}(F)$, which satisfy invariant $\mathbf{I}_1$–$\mathbf{I}_4$. We maintain the regions $C(F)$, $F \in \mathcal{F}$, in the three phases of Algorithm 1 as follows:
Phase 1. In case 1, a new edge uv decomposes the face F_u into two faces, whose deformable edges are $\tau(F_u)$ and uv, according to Rule (∗). We split $C(F_u) = \mathrm{int}(F_u)$ into two regions along uv. At the end of phase 1, we still have $\mathcal{C} = \mathcal{F}$. It is easily checked that invariants $\mathbf{I}_1$–$\mathbf{I}_4$ still hold.

Phase 2. By Lemma 6.2, no deformable edges has been deformed yet. In Case 2, we have a vertex $u \in R$ such that $\vec{a}_u$ hits an edge $vw \neq \tau(F_u)$ at $x_u = vw \cap \vec{a}_u$. The shortest path $\mathrm{path}(u, w) = (u = p_0, p_1, \ldots, p_\ell = w)$ is homotopy equivalent to the path (u, x_u, w). A pseudo-triangle is formed by $\mathrm{path}(u, w)$ and the segments ux_u and $x_u w$, and so $\mathrm{path}(u, w)$ passes through reflex vertices of F_u at $p_1, p_2, \ldots, p_{\ell-1}$. Each new deformable edge $p_j p_{j+1}$ is adjacent to a unique new face in $\mathcal{F}$.

By $\mathbf{I}_2$, the initial portion of $\vec{a}_u$ is in $C(F_u)$. Let y_u be the point where $\vec{a}_u$ hits the boundary of $C(F_u)$. Refer to Fig. 10. By invariant $\mathbf{I}_4$ and because $\vec{a}_u$ does not hit $\tau(F_u)$, ray $\vec{a}_u$ hits the boundary of $C(F_u)$ at or before x_u. That is, y_u lies on the segment ux_u. If $y_u = x_u$, then we decompose $C(F_u)$ as follows. The rays $\vec{a}_{p_j}$, $j = 0, 1, \ldots, \ell - 1$ (in this order) split $C(F_u)$ into $\ell + 1$ regions. Each new deformable edge $p_j p_{j+1}$ lies between two bisectors, $\vec{a}_{p_j}$ and $\vec{a}_{p_{j+1}}$, and hence in a unique new region (Fig. 10, first row). It is easy to check that invariants $\mathbf{I}_1$–$\mathbf{I}_4$ are maintained.

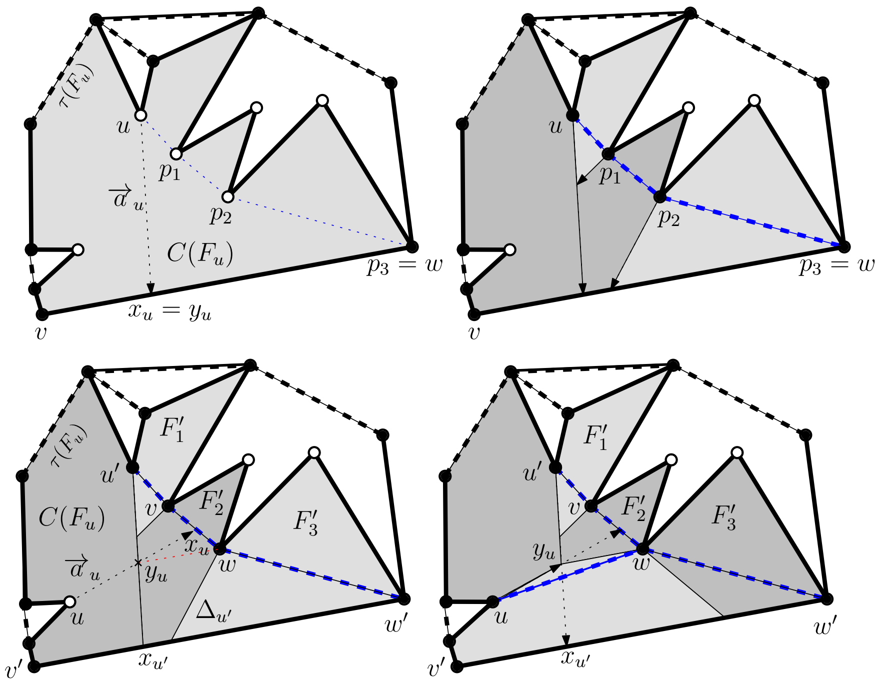

Figure 10: Decomposing a cell $C(F_u)$ along the rays $\vec{a}_u$ in two consecutive steps in phase 2. First row: $x_u = y_u$, and $C(F_u)$ is partitioned along $\vec{a}_{p_j}$, $j = 0, 1, 2$. Second row: $x_u \neq y_u$, the regions associated with F'_1, F'_2 and F'_3 spill over into F_u, and some of them are truncated along $y_u w$ so that they are disjoint from any newly inserted edge.

If $y_u \neq x_u$, then we need to be more careful because the regions $C(F')$ of some faces $F' \in \mathcal{F}$ adjacent to F extends into the interior of F_u (Fig. 10, second row). In order to maintain invariant $\mathbf{I}_4$ for such a face F', the region $C(F')$ may have to be truncated to assure that it is disjoint from the new edges $p_j p_{j+1}$, $j = 0, 1, \ldots, \ell - 1$. Point y_u lies on the ray $\vec{a}_{u'}$ emitted from some vertex u' processed earlier in phase 2. By the choice of reflex vertices in phase 2, u' and u have opposite orientations with respect to $\tau(F_u)$. Assume, without loss of generality, that u is ccw and u' is cw. The part of face F_u on the left side of $\vec{a}_{u'}$ is a pseudo-triangle $\Delta_{u'}$ bounded by segments $u'x_{u'}$, $x_{u'}v_{u'}$, and a reflex chain $(u' = p'_0, p'_1, \ldots, p'_{\ell'} = w')$; and $\Delta_{u'}$ is covered by regions associated with the faces adjacent to $p'_0 p'_1, \ldots, p'_{\ell'-1} p'_{\ell'}$. We define an auxiliary region $\tilde{C}_u$ to be the union of $C(F_u)$ and part of $\Delta_{u'}$. Specifically, decompose $\Delta_{u'}$ along segment $y_u w$ and ray $\overrightarrow{vw}$, and let $\tilde{C}_u$ contain the part lying on the same side as uw. The region $\tilde{C}_u$ is the union of $C(F_u)$ and a convex set adjacent to $y_u w$. Hence, the reflex vertices of $\tilde{C}_u$ are y_u and some reflex vertices of F_u in R (including u). Now the rays $\vec{a}_{p_j}$, $j = 0, 1, \ldots, \ell - 1$ (in this order) split region $\tilde{C}_u$ into $\ell + 1$ regions. Each new deformable edge $p_j p_{j+1}$ lies between two bisectors, $\vec{a}_{p_j}$ and $\vec{a}_{p_{j+1}}$, and hence in a unique new region. Since $\vec{a}_u$ passes through y_u, the resulting regions in $\mathcal{C}$ do not have reflex vertices at y_u. The regions covering $\Delta_{u'}$ have been truncated such that they are disjoint from all new edges, and they have no reflex vertices in $\Delta_{u'}$. Invariants $\mathbf{I}_2$–$\mathbf{I}_4$ are now maintained for all faces $F \in \mathcal{F}$.

Phase 3. For every $u \in R$, ray $\vec{a}_u$ hits edge $\tau(F_u) = vw$, and vw is replaced by edges uv and uw. Let $\vec{a}_u$ split region $C(F_u)$ into two regions. By invariant $\mathbf{I}_4$, $vw = \tau(F_u)$ is either adjacent to or lies in cell $C(F_u)$, and cell $C(F_u)$ may extend to the opposite side of $\tau(F_u)$. Since u is the vertex of R closest to the line spanned by $\tau(F_u)$, and all reflex angles of F_u with apex in R are reflex angles of $C(F_u)$ (invariant $\mathbf{I}_3$), we conclude that the relative interiors of both uv and uw are in $C(F_u)$, and hence in F_u. Now it is easy to verify that invariants $\mathbf{I}_2$–$\mathbf{I}_4$ are maintained in this case, too. $\square$

Corollary 6.7 *When Algorithm 1 has processed all vertices in R, every region in $\mathcal{C}$ is convex, and $\mathcal{C}$ is a tiling of* $\mathrm{ch}(G)$ *into convex regions. The convex cells in $\mathcal{C}$ have property (♡), where a convex cell $C(F)$ corresponds to the deformable edge $\tau(F)$.* $\square$

Stage 6. Connecting singletons. There are s_i singletons in the interior of $\mathrm{ch}(G)$, which lie in convex regions $C_j \in \mathcal{C}$ with property (♡). In each convex region C_j, $j = 1, 2, \ldots, \ell$, we replace the deformable edge $e_j = u_j v_j$ by a path between u_j and v_j that lies entirely in C_j and passes through all singletons in C_j. Let m be the number of singletons in the interior of C_j. Label them as $p_1, p_2, \ldots, p_m$ as follows. First label the singletons on the left of $\overrightarrow{u_j v_j}$ in the decreasing order of angles $\angle(v_j, u_j, p_i)$; then label the singletons on the right of $\overrightarrow{u_j v_j}$ in increasing order of angles $\angle(u_j, v_j, p_i)$. See two examples in Fig. 11. Replace edge $u_j v_j$ by the simple path $(u_j, p_1, p_2, \ldots, p_m, v_j)$. Replace any remaining pairs of parallel edges by a single edge. We obtain a 2-edge-connected PSLMG G_6. The number of edges has increased by $\boxed{s_i}$. Each of the s_i singletons of G_5 becomes a 3-edge-connected component in G_6. Hence the number of 3-edge-connected components does not change, we have $\lambda(G_6) = \lambda(G_5) \leq c_h + g_h + s$.

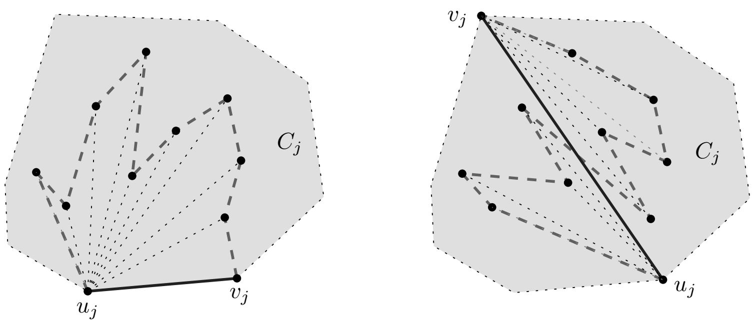

Figure 11: Expanding a deformable edge $u_j v_v$ to a path.

Stage 7. Eliminating 2-bridges. The input PSLG G was 3-edge-augmentable. In stages 1-6, we have not added any chords of $\mathrm{ch}(G)$, and so at the end of stage 6, we have a 2-edge-connected 3-edge-augmentable PSLMG G_6. We apply Lemma 5.3 to obtain a 3-edge-connected PSLG G_7 with at most $\lambda(G_6) - 1 = \boxed{c_h + g_h + s - 1}$ new edges. This completes our augmentation algorithm.

Theorem 2 *Every 3-edge-augmentable* PSLG *G with $n \geq 4$ vertices can be augmented to 3-edge-connected* PSLG *with at most $2n - 2$ new edges.*

Proof. In stages 1-7, we have added at most $b_i + c + f + r + 2s - 1$ new edges. If G is not a forest, this is at most $2n - 2$ by Corollary 5.5. If G is a forest and has an edge (bridge) along $\mathrm{ch}(G)$, then $b_h \geq 1$, and again $b_i + c + f + r + 2s - 1 \leq 2n - 2$.

Let G be a forest with no edges along the convex hull (i.e., $b_h = 0$). We would like to show that our augmentation algorithm used fewer than $b_i + c + f + r + 2s - 1$ new edges. We distinguish four cases.

• *Case 1. A non-singleton component of G has two vertices, u and v, along the convex hull.* By adding all hull edges in stage 2, we eliminate the bridges along the path between u and v, and so we add fewer than b_i new edges in stage 4.

• *Case 2. Every component of G has at most one vertex along the convex hull, and $c_i \geq 1$.* Then in stage 3 we add two new edges to a vertex of each non-singleton component in the interior of $\mathrm{ch}(G)$. The two edges form a circuit with K, and so it either decreases $\lambda_h(G_3)$ or eliminates a bridge in the interior of $\mathrm{ch}(G)$.

• *Case 3. Every component of G has at most one vertex along the convex hull, $c_i = 0$, but G_2 has a bridge.* Then in stage 4 we add a new edge for each bridge. The first such new edge creates a circuit, which either contains another bridge of G_3 (eliminating at least two bridges at once), or contains a hull edge, decreasing $\lambda_h(G_3)$ by at least one.

• *Case 4. G consists of n singletons.* Then one can augment G to a plane Hamiltonian circuit H (e.g., the Euclidean TSP tour on n vertices), with n new edges. The circuit H is 3-augmentable by Proposition 2.2. We can augment the edge-connectivity from 2 to 3 with at most $n-2$ new edges by Theorem 1. In this case, again, G can be augmented to a 3-edge-connected PSLG with at most $2n - 2$ new edges. □

7 Algorithmic aspects

In this section, we study the algorithmic aspects of our results. The input of all our algorithms is a PSLG G with $n \geq$ vertices, no three of which are collinear. Note that the non-collinearity condition is not necessary for our results in Sections 2–6. In this section, however, we rely on data structures that are available only under this additional condition. (It is likely that all our algorithms can be implemented with the same time bounds for arbitrary vertex sets, but the possible generalization of the relevant data structures goes beyond the scope of this paper.)

We use the real RAM model of computation. We assume that the input PSLG G is given with the coordinates of all vertices and an edge list. We store each connected component of a PSLG in the standard *doubly-connected edge list* (for short, *DCEL*) data structure, which can be constructed in $O(n)$ time [32]. A PSLG with n vertices has $O(n)$ edges. The DCEL maintains a vertex list, a face list, and then stores, for each edge, the two incident vertices, the adjacent faces on the two sides of the edge (a bridge edge is adjacent to the same face on both sides), and the counterclockwise next edge in each face. From the DCEL, one can easily extract the boundary of each face (in a connected component), and the adjacent edges of each vertex in the order in which they appear around the vertex.

Testing augmentability. Given a PSLG G with $n \geq 4$ vertices, we can test in $O(n \log n)$ time whether G is 3-augmentable or 3-edge-augmentable. We apply Propositions 2.2 and 2.3. Compute the convex hull $\mathrm{ch}(G)$ in $O(n \log n)$ time [11], mark each extreme vertex in the vertex list, and then mark the edges of G that join two nonconsecutive extreme vertices as chords of $\mathrm{ch}(G)$. If not all vertices lie on the convex hull and none of the edges is a chord of $\mathrm{ch}(G)$, then G is 3-augmentable.

For testing whether G is 3-edge-augmentable, we need to check for each chord edge of $\mathrm{ch}(G)$ whether the vertices lying on either side are all on the convex hull. The chord edges partition $\mathrm{ch}(G)$ into convex *sectors*. We can construct a *dual graph* on the sectors in $O(n)$ time: the nodes corresponds to sectors, two nodes are adjacent iff the corresponding sectors are adjacent to the same chord of $\mathrm{ch}(G)$. The dual graph is a tree, since the removal of each chord disconnected $\mathrm{ch}(G)$. Now G is 3-edge-augmentable if and only if each sector corresponding to a leaf in the dual graph contains a vertex of G in its interior. Compute the boundaries of the sectors corresponding to the leaves in the dual graph in $O(n)$ time. A sweep-line algorithm can detect whether each leaf sector contains a vertex lying in their interior in $O(n \log n)$ time.

If G is connected then the time complexity drops to $O(n)$. The convex hull $\mathrm{ch}(G)$ can be computed in $O(n)$ time [24]. Instead of a sweep-line algorithm, we can traverse the part of G clipped in each leaf sector in $O(n)$ total time, and report whether each contains a vertex in their interior.

Constructing 3-edge-connected or 3-connected triangulations. Every 3-edge-augmentable PSLG G with $n \geq 4$ vertices can be augmented to a 3-edge-connected triangulation in $O(n \log n)$ time. If G is connected then the time complexity drops to $O(n)$. We follow the algorithm described in the proof of Proposition 2.3. Compute the convex hull $\mathrm{ch}(G)$ in $O(n \log n)$ time ($O(n)$ time if G is connected), and augment G with all hull edges. If G is disconnected, then scan G with a vertical sweep line in $O(n \log n)$ time, and in every component lying in the interior of $\mathrm{ch}(G)$, connect the right-most vertex to an arbitrary visible vertex on its right. The result is a *connected* 3-edge-augmentable PSLG G_0.

Every bounded face of G_0 is a weakly simple polygon. We triangulate each bounded face independently. If G_0 is 3-augmentable, then every bounded face has a vertex that lies in the interior of $\mathrm{ch}(G)$. Let P be a face of G_0, and assume that P has m vertices. If P has a vertex v that lies in the interior of $\mathrm{ch}(G)$, then augment G' with the shortest paths from v to all other vertices of P. All shortest paths can be computed in $O(m)$ time [12], and they decompose P into pseudo-triangles. Every diagonal in a pseudo-triangle is incident to a reflex vertex, and a reflex vertex necessarily lies in the interior of $\mathrm{ch}(G)$. So we can arbitrarily triangulate every pseudo-triangle to obtain a triangulation of P.

If G is 3-edge-augmentable, then it is possible that every vertex of P lies on $\mathrm{ch}(G)$. Then P is a convex polygon, and it has two edges, say e and f, which are chords of $\mathrm{ch}(G)$. Denote the vertices of P by $u_1, u_2, \ldots, u_m$ in counterclockwise order, such that $e = u_1u_2$ and $f = u_ju_{j+1}$. Triangulate P by adding the diagonals $u_1u_3, u_1u_4, \ldots, u_1u_j$ and $u_ju_{j+2}, u_ju_{j+3}, \ldots, u_ju_m$. All new diagonals separate e and f. Since we have not added any chord such that all vertices on one side lie on the convex hull, the resulting triangulation is 3-edge-connected.

Augmenting the edge-connectivity from two to three. Given a 2-edge-connected and 3-edge-augmentable PSLG $G = (V, E)$ with $n \geq 4$ vertices, we can augment G it to a 3-edge-connected PSLG in $O(n\alpha(n))$ time by adding at most $\lambda(G) - 1$ new edges, where $\alpha(n)$ is the inverse Ackermann function (which grows extremely slowly). We follow the algorithm in the proof of Lemma 5.3. Construct a 3-edge-connected triangulation $G' = (V, E')$ for G in $O(n)$ time (as above). La Poutré [31] devised a semi-dynamic data structure for maintaining the 3-edge-connected components of a graph. Starting from the empty graph, it supports $O(n)$ edge insertions in $O(n\alpha(n))$ total time. It can answer queries, whether two vertices are in the same 3-edge-connected component, in $O(1)$ time. (For an $O(n \log n)$ time algorithm, we may use simpler data structures in [30, 7], which runs in $O(n \log n)$ total time.)

Augmenting the edge-connectivity to three. Given a 3-edge-augmentable PSLG G with $n \geq 4$ vertices, it can be augmented to a 3-edge-connected PSLG in $O(n \log n)$ time by adding at most $2n - 2$ new edges.

We store each connected component of a PSLG in a standard DCEL data structure. We also maintain the boundary of each face and the neighbors of each vertex (i.e., the rotation of the vertex) in doubly-linked lists, endowed with binary search trees. In the component $K \subseteq G$ containing the convex hull vertices, we also maintain a dynamic data structure of Goodrich and Tamassia [10]. For a connected PSLG with n vertices, it requires $O(n)$ space and $O(n \log n)$ preprocessing time. It supports any ray shooting query, edge insertion, and deletion in $O(\log^2 n)$ time. Between any two points, it can report the length of shortest path, as well as the first and last edges of the shortest path in $O(\log^2 n)$ time. Note that this data structure works for connected graphs only (each face has to be simply connected), so it is important that we maintain it only for *one* component of G (ray shooting among disjoint components would require heavier machinery).

We consider the seven stages of our augmentation algorithm one by one. In stage 1, we can easily choose a *deformable* edge in each bounded face. The DCEL data structure supports the maintenance of a deformable edge for the bounded faces in stages 2-5. When a new edge splits a face F into two subfaces we can tell which subface is bounded and which is adjacent to $\tau(F)$ in $O(\log n)$ time. So we can assign a deformable edge to each bounded subface in $O(\log n)$ time. Since the number of faces remains $O(n)$ during the augmentation, the deformable edges can be maintained in $O(n \log n)$ total time. In stage 2, the convex hull $\mathrm{ch}(G)$ can be computed in $O(n \log n)$ time.

For the recursive procedure in stage 3, we also maintain a semi-dynamic (delete-only) convex hull data structure for the set U of vertices of the subgraph H of all non-singleton components lying in the interior of $\mathrm{ch}(G)$. Data structures by Chazelle [2] and by Hershberger and Suri [14] support $O(n)$ vertex deletions in $O(n \log n)$ total time. For a vertex $u \in U$, we can find the first edges uv_1, uw_q of the shortest path from u to v and w, respectively, in $O(\log^2 n)$ time with the data structure of Goodrich and Tamassia [10]. We find the vertices u_1 with a tangent queries to the convex hull $\mathrm{ch}(H)$, and use binary search for finding adjacent vertices u' and u'' along $\mathrm{ch}(H)$.

In stage 4, we are given a connected PSLG G_3 (ignoring the singletons for the moment), and we augment it to a 2-edge-connected PSLG with at most b new edges. We maintain a union-find data structure for the 2-edge-connected components of G_3. Compute a 3-edge-connected triangulation T containing G_3 in $O(n)$ time (as described above). We augment G_3 incrementally as follows. Augment G_3 with every edge e in $T - G_3$ which connects distinct 2-edge-connected components of the current graph. The addition of any new edge merges two 2-edge-connected components into one, so we augment G_3 with exactly b new edges.

In Stage 5, Algorithm 1 can be implemented in $O(n \log^2 n)$ time using $O(n)$ ray shooting and shortest paths queries of the Goodrich-Tamassia data structure. Similarly, the construction of the subdivision $\mathcal{C}$ (described in Lemma 6.6) can be implemented with $O(n)$ ray shooting queries of a separate Goodrich-Tamassia data structure, for the subdivision $\mathcal{C}$ of $\mathrm{ch}(G)$. In stage 6, all deformable edges can be deformed to paths visiting all singletons lying in the corresponding convex region in $O(n \log n)$ total time. Finally, Stage 7 can be completed in $O(n\alpha(n))$ time, as described above, where $\alpha(.)$ is the inverse Ackermann function.

8 Conclusion

We have described how to augment a 3-edge-augmentable PSLG with $n \geq 4$ vertices to a 3-edge-connected PSLG using at most $2n - 2$ new edges. The resulting graph is not necessarily 3-connected (e.g., if the input is not 3-augmentable). It remains an open problem whether any 3-augmentable PSLG with $n \geq 4$ vertices can be augmented to a 3-connected PSLG with at most $2n - 2$ new edges.

Conjecture 8.1 *Every 3-augmentable* PSLG *with* $n \geq 4$ *vertices can be augmented to a 3-connected* PSLG *by adding at most* $2n - 2$ *new edges.*

We may consider embedding preserving augmentations of a PSLG such that the new edges are allowed to be arbitrary Jordan arcs (not necessarily straight line segments). In this case, the input is a PSLG and the output is a *planar topological graph* (for short, PTG). If the new edges are not restricted to be straight line segments, then the constraints in Proposition 2.1 and Proposition 2.2 no longer apply. Every PSLG with $n \geq 4$ vertices has an embedding preserving augmentation to a 3-connected and 3-edge-connected PTG: Any PSLG can be triangulated with curved edges such that the outer face is a triangle. Determining the minimum number of new edges that can augment any PSLG with $n \geq 4$ vertices to a 3-connected or 3-edge-connected PTG is an open problem.

Conjecture 8.2 *Every 2-connected* PSLG G *with* $n \geq 4$ *vertices can be augmented to a 3-connected topological graph by adding at most* $\frac{4}{5}n - O(n)$ *new edges.*

Conjecture 8.3 *Every 2-edge-connected* PSLG G *with* $n \geq 4$ *vertices can be augmented to a 3-edge-connected topological graph by adding at most* $\frac{4}{5}n - O(n)$ *new edges.*

These bounds would be optimal in the worst case. A lower bound construction consists of a (straight line) triangulation on k vertices, which has $2k - 4$ faces, and a path with two internal vertices lying in every face. So G has a total of $n = k+(2k-4)2 = 5k-8$ vertices, and we have $k = (n+8)/5$. In order to raise the degree of every vertex in the interior of the triangular faces to 3, we need at least $(2k-4)2 = \frac{4}{5}(n-2)$ new edges. This is also a lower bound on the number of curved edges required for augmenting the connectivity or edge-connectivity to three.

Finally, note that all our results are only worst-case optimal. An arbitrary 3-edge-augmentable PSLG G may be augmented to 3-edge-connectivity with significantly fewer than $2n-2$ new edges. Our augmentation algorithm may add a large number of edges (e.g., one new edge for each reflex vertex in Stage 5) even if a single new edge would be sufficient. It would be interesting to design efficient approximation algorithms for the edge-connectivity and vertex-connectivity augmentation of PSLGs.

References

[1] M. Abellanas, A. García, F. Hurtado, J. Tejel, and J. Urrutia, Augmenting the connectivity of geometric graphs, *Comput. Geom. Theory Appl.* **40** (3) (2008), 220–230.

[2] B. Chazelle, On the convex layers of a planar set, *IEEE Trans. Inf. Theory* **IT-31** (4) (1985), 509–517.

[3] E. Cheng and T. Jordán, Successive edge-connectivity augmentation problems, *Math. Program.* **84** (1999), 577-593.

[4] K. P. Eswaran and R. E. Tarjan, Augmentation problems, *SIAM J. Comput.* **5** (4) (1976), 653-665.

[5] S. Fialko and P. Mutzel, A new approximation algorithm for the planar augmentation problem in *Proc. 9th ACM-SIAM Sympos. on Discrete Algorithms*, ACM Press, 1998, pp. 260–269.

[6] A. Frank, Augmenting graphs to meet edge-connectivity requirements, *SIAM J. Discrete Math.* **5** (1) (1992), 22–53.

[7] Z. Galil and G. F. Italiano, Maintaining the 3-edge-connected components of a graph on-line, *SIAM J. Comput.* **22** (1) (1993), 11–28.

[8] A. García, F. Hurtado, C. Huemer, J. Tejel, and P. Valtr, On triconnected and cubic plane graphs on given point sets, *Computational Geometry: Theory and Applications* **42** (2009), 913–922.

[9] A. García, F. Hurtado, M. Noy, and J. Tejel, Augmenting the connectivity of outerplanar graphs, *Algorithmica* **56** (2) (2010), 160–179.

[10] M. T. Goodrich, R. Tamassia, Dynamic ray shooting and shortest paths in planar subdivisions via balanced geodesic triangulations, *J. Algorithms* **23** (1997), 51–73.

[11] R. L. Graham, An efficient algorithm for determining the convex hull of a finite planar set, *Inf. Proc. Letts.* **1** (1972), 132–133.

[12] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan, Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons, *Algorithmica* **2** (1987), 209–233

[13] C. Gutwenger, P. Mutzel, and B. Zey, On the hardness and approximability of planar biconnectivity augmentation, in *Proc. COCOON*, vol. 5609 of LNCS, Springer, 2009, pp. 249-257.

[14] J. Hershberger and S. Suri, Applications of a semi-dynamic convex hull algorithm, *BIT* **32** (1992), 249–267.

[15] T.-S. Hsu and V. Ramachandran, On finding a minimum augmentation to biconnect a graph, *SIAM J. Computing*, **22** (5) (1993), 889–912.

[16] T.-S. Hsu, Simpler and faster biconnectivity augmentation, *J. Algorithms* **45** (1) (2002), 55–71.

[17] B. Jackson and T. Jordán, Independence free graphs and vertex connectivity augmentation, *J. Comb. Theory Ser. B* **94** (2005), 31-77.

[18] G. Kant, Augmenting outerplanar graphs, *J. Algorithms* **21** (1996), 1-25.

[19] G. Kant and H. L. Bodlaender, Planar graph augmentation problems, in: *Proc. 2nd Workshop on Algorithms and Data Structures*, vol. 519 of LNCS, Springer, 1991, pp. 286–298.

[20] G. Kortsarz and Z. Nutov, Approximating minimum cost connectivity problems, Chap. 58 in *Handbook of Approximation Algorithms and Metaheuristics (T. F. Gonzalez, ed.)*, CRC Press, Boca Raton, 2007.

[21] D. R. Karger and D. Panigrahi, A near-linear time algorithm for constructing a cactus representation of minimum cuts, in *Proc. 19th ACM-SIAM Sympos. Discrete Alg.*, ACM Press, 2009, pp. 246–255

[22] L. Lovász, *Combinatorial Problems and Exercises*, North-Holland, Amsterdam, 1979.

[23] W. Mader, A reduction method for edge-connectivity in graphs, *Ann. Discr. Math.* **3** (1978), 145–164.

[24] D. McCallum and D. Avis, A linear algorithm for finding the convex hull of a simple polygon, *Inf. Proc. Letts.* **9** (5) (1979), 201.

[25] H. Nagamochi and P. Eades, An edge-splitting algorithm in planar graphs, *J. Combin. Opt.* **7** (2) (2003), 137-159.

[26] H. Nagamochi and T. Ibaraki, Augmenting edge-connectivity over the entire range in $\tilde{O}(nm)$ time, *J. Algorithms* **30** (1999), 253-301.

[27] H. Nagamochi and T. Ibaraki, Graph connectivity and its augmentation: applications of MA orderings, *Discrete Appl. Math.* **123** (2002), 447-472.

[28] H. Nagamochi, Y. Nakao and T. Ibaraki, A fast algorithm for cactus representations of minimum cuts, *Japan J. Ind. Appl. Math.* **17** (2) (2000), pp. 245-264

[29] J. Plesník, Minimum block containing a given graph, *Arch. Math.* **27** (6) (1976), 668–672.

[30] J. A. La Poutré, J. van Leeuwen and M. H. Overmars, Maintenance of 2- and 3-edge-connected components of graphs I, *Discrete Maths.* **114** (1993), 329–359.

[31] J. A. La Poutré, Maintenance of 2- and 3-edge-connected components of graphs II, *SIAM J. Comput* **29** (2000), 1521–1549.

[32] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Texts and Monographs in Computer Science, 1985, Springer.

[33] G. Reinelt, D. O. Theis, and K. M. Wenger, Computing finest mincut partitions of a graph and application to routing problems, *Discrete Appl. Math.* **156** (3) (2008), 385–396.

[34] A. Rosenthal and A. Goldner, Smallest augmentations to biconnect a graph, *SIAM J. Computing* **6** (1977), 55–66.

[35] I. Rutter and A. Wolff, Augmenting the connectivity of planar and geometric graphs, *Proc. Conf. Topological Geom. Graph Theory (Paris, 2008)*, pp. 55–58.

[36] D. L. Souvaine and Cs. D. Tóth, A vertex-face assignment for plane graphs, *Comput. Geom. Theory Appl.* **42** (5) (2009), 388–394.

[37] Cs. D. Tóth, Connectivity augmentation in planar straight line graphs, *European Journal of Combinatorics* (2010), to appear. `http://math.ucalgary.ca/~cdtoth/2edgecon20.pdf`

[38] W. T. Tutte, *Connectivity in Graphs*, University of Toronto Press, 1966.

[39] L. Végh, Augmenting undirected node-connectivity by one, in *STOC*, 2010, ACM Press pp. 563–572.

[40] T. Watanabe and A. Nakamura, Edge-connectivity augmentation problems, *Comp. Sys. Sci.* **35** (1) (1987), 96–144.