

The Price of Connectivity Augmentation on Planar Graphs

Hugo A. Akitaya 

University of Massachusetts Lowell, MA, USA

Justin Dallant 

Aarhus University, Denmark

Erik D. Demaine 

CSAIL, Massachusetts Institute of Technology,
Cambridge, MA, USA

Michael Kaufmann 

Wilhelm-Schickard Institut für Informatik,
University of Tübingen, Germany

Linda Kleist 

Department of Informatics, Universität Hamburg,
Germany

Frederick Stock 

University of Massachusetts Lowell, MA, USA

Csaba D. Tóth 

Department of Mathematics, California State
University Northridge, Los Angeles, CA, USA
Department of Computer Science,
Tufts University, Medford, MA, USA

Torsten Ueckerdt 

Karlsruhe Institute of Technology, Germany

Abstract

Given two classes of graphs, $\mathcal{G}_1 \subseteq \mathcal{G}_2$, and a c -connected graph $G \in \mathcal{G}_1$, we wish to augment G with a smallest cardinality set of new edges F to obtain a k -connected graph $G' = (V, E \cup F) \in \mathcal{G}_2$. In general, this is the $c \rightarrow k$ connectivity augmentation problem. Previous research considered variants where $\mathcal{G}_1 = \mathcal{G}_2$ is the class of planar graphs, plane graphs, or planar straight-line graphs. In all three settings, we prove that the $c \rightarrow k$ augmentation problem is NP-complete when $2 \leq c < k \leq 5$.

However, the connectivity of the augmented graph G' is at most 5 if $\mathcal{G}_2$ is limited to planar graphs. We initiate the study of the $c \rightarrow k$ connectivity augmentation problem for arbitrary $k \in \mathbb{N}$, where $\mathcal{G}_1$ is the class of planar graphs, plane graphs, or planar straight-line graphs, and $\mathcal{G}_2$ is a beyond-planar class of graphs: ℓ -planar, ℓ -plane topological, or ℓ -plane geometric graphs. We obtain tight bounds on the tradeoffs between the desired connectivity k and the local crossing number ℓ of the augmented graph G' . We also show that our hardness results apply to this setting.

The connectivity augmentation problem for triangulations is intimately related to edge flips; and the minimum augmentation problem to the flip distance between triangulations. We prove that it is NP-complete to find the minimum flip distance between a given triangulation and a 4-connected triangulation, settling an open problem posed in 2014, and present an EPTAS for this problem.

2012 ACM Subject Classification Mathematics of computing $\rightarrow$ Paths and connectivity problems; Mathematics of computing $\rightarrow$ Graph algorithms; Theory of computation $\rightarrow$ Computational geometry

Keywords and phrases connectivity augmentation, local crossing number, flip distance

Digital Object Identifier 10.4230/LIPIcs.GD.2025.23

Related Version A full version will appear on arXiv shortly before Graph Drawing 2025.

Full Version: <https://arxiv.org/abs/2509.01096>

Funding *Hugo A. Akitaya*: Research supported in part by NSF grant CCF-2348067.

Csaba D. Tóth: Research supported in part by the NSF award DMS-2154347.

Torsten Ueckerdt: Research supported by the German Research Foundation DFG-520723789.

Acknowledgements The work in this paper originated at the 11th Annual Workshop on Geometry and Graphs held at the Bellairs Research Institute, 8–15 March, 2024.



© Hugo A. Akitaya, Justin Dallant, Erik D. Demaine, Michael Kaufmann, Linda Kleist, Frederick Stock, Csaba D. Tóth, and Torsten Ueckerdt;
licensed under Creative Commons License CC-BY 4.0

33rd International Symposium on Graph Drawing and Network Visualization (GD 2025).

Editors: Vida Dujmović and Fabrizio Montecchiani; Article No. 23; pp. 23:1–23:24

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Connectivity augmentation is a classical problem in combinatorial optimization [21, 34]: Given a c -connected graph $G = (V, E)$ and a positive integer k , find a minimum set F of new edges such that the graph $G' = (V, E \cup F)$ is k -connected. For undirected and unweighted graphs, Jackson and Jordán [32] gave a polynomial-time algorithm for any c and k (previous results addressed $c = 2, 3, 4$ [19, 29, 55]; see also [53]). For the weighted version, finding a minimum-cost augmentation is NP-complete [22], APX-hard, and the best approximation ratio is 1.5 [51]. In the remainder of this paper, we consider unweighted undirected graphs.

We investigate connectivity augmentation over planar graphs in three different settings, Figure 1 depicts an illustrative example: In the *abstract graph setting*, G is planar, and G' must also be planar. For this variant, NP-completeness [37] and a $\frac{5}{3}$ -approximation are known for $k = 2$ [20, 27]. Moreover, there exists an $O(n^3)$ -time $\frac{5}{4}$ -approximation algorithm for $c = 2$ and $k = 3$ [37]. In the *topological graph setting*, G and G' are plane graphs (with a fixed embedding) and G' extends the given embedding: the problem remains NP-complete for $c = 0$ and $k = 2$, but there is a linear-time algorithm for $c = 1$ and $k = 2$ [27, 28]. Finally, in the *geometric graph setting*, both G and G' are planar straight-line graphs (PSLG, for short), and G' extends the given embedding [1, 4, 6, 31, 47]. The three settings reveal deep structural properties of embedded graphs as the edges of the input graph G form topological or geometric obstructions to possible new edges.

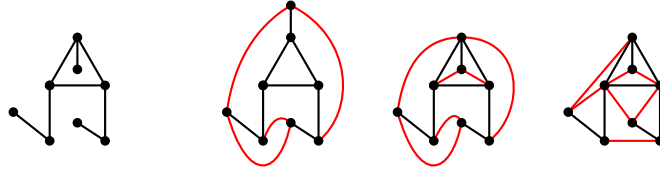


Figure 1 A planar straight-line graph, and its 3-connected minimum augmentation in the abstract, topological, and geometric settings with 4, 6, and 7 additional edges, respectively.

However, planarity severely limits the feasible values of k : Every planar graph has a vertex of degree at most 5, and hence augmentation is infeasible for any $k > 5$. Furthermore, for $n \geq 3$ points in convex position, every PSLG has a vertex of degree at most 2. The connectivity augmentation problem for PSLGs is already challenging for $k = 1, 2$; and requires additional assumptions about the point configurations for $k = 3$; see [25].

In this paper, we also consider beyond-planar variants of the problem: The augmented graph G' need not be planar but “close” to planarity. In particular, we study the trade-offs between connectivity and local crossing number, defined as follows. An abstract graph G is ℓ -planar if it admits a drawing in the plane such that each edge has at most ℓ crossings. The *local crossing number* of G , denoted $\text{lcr}(G)$, is the minimum integer ℓ such that G is ℓ -planar.

We also consider connectivity augmentation restricted to *topological graphs* and *geometric graphs*. In both cases, V is a set of n points in the plane in general position (i.e., no three points are collinear), the edges are Jordan arcs in topological graphs and straight-line segments in geometric graphs. In both cases, the drawing of G is fixed, and the *local crossing number* is defined as the maximum number of crossings per edge *in that drawing*.

Asymptotic Bounds and Tradeoffs. We start with an asymptotically tight tradeoff between connectivity and local crossing number (regardless of the number of new edges): Every k -connected graph is $O(k^2)$ -planar (Proposition 2.2). For geometric graphs on n points in

convex position, we determine the precise tradeoff (Proposition 2.3). The asymptotically tight bound can be attained even if we augment a given plane triangulation in the topological setting (Theorem 3.2). The geometric setting turns out to be much more constrained: We show that any PSLG can be augmented to a 3-connected 5-planar geometric graph, and this bound is the best possible (Theorem 8.1). However, augmenting a PSLG to a 4-connected geometric graph may already increase the local crossing number to $\Omega(n)$ (Theorem 8.3).

Algorithmic Results. Table 1 presents an overview of the computational complexity of finding minimum solutions; each field has an entry for the abstract, topological, and geometric setting (in this order); a single entry means that the same result holds in all three settings.

■ **Table 1** Summary of complexity results. Bold results are proved in this paper. Arrows indicate that a result is implied by the one pointed to.

From/To	1	2	3	4	5
Tree	—	P [35],[27],[4]	P [35], P , ?	?	?
0	P	NPC [37],[28],[46]	NPC [47]	NPC [47]	NPC [47]
1	—	NPC[37], P[27], P[4]	NPC[37],[47],[4]	NPC[37],[47],[4]	NPC[37],[47],[4]
2	—	—	NPC	NPC (↓)	NPC (↓)
3	—	—	—	NPC	NPC (↓)
4	—	—	—	—	NPC

Trees. Kant [35] gave an $O(n)$ -time algorithm for the 2- and 3-connectivity augmentation of outerplanar graphs with n vertices in the abstract setting, and Gutwenger et al. [27] gave an $O(n(1 + \alpha(n)))$ -time algorithm for the $1 \rightarrow 2$ augmentation for connected plane graphs. For a PSLG tree on n vertices, there is an $O(n^4)$ -time algorithm for 2-connectivity augmentation [4]. (Biconnectivity augmentation can also be solved optimally over outerplanar graphs, where both G and G' must be outerplanar [26].) We give an $O(n)$ -time algorithm for the minimum augmentation of a plane tree to a 3-connected plane graph in the topological settings (Theorem 4.2). In the geometric setting, 3-connectivity augmentation of trees remains open.

Geometric Triangulations. For a straight-line triangulation on n points in convex position, we give a combinatorial characterization of 3- and 4-connected augmentations. In this setting, the dual graph is a tree. We use the dual tree for a dynamic programming algorithm to compute a minimum augmentation to a k -connected ℓ -planar geometric graph for $k \in \{3, 4\}$ and constant $\ell \in \mathbb{N}$ in $O(n)$ time (Theorem 7.2).

Hardness. In most settings, however, the $c \rightarrow k$ connectivity augmentation problem is NP-complete. We prove the following theorem using several reductions from planar 3-SAT.

► **Theorem 1.1.** *Given a c -connected planar graph G and an integer τ , deciding whether there is an edge set of cardinality at most τ that augments G into a planar k -connected G' is NP-complete when $c \in \{2, 3, 4\}$, $k \in \{3, 4, 5\}$, and $c < k$. The problem remains NP-complete if G is a topological or geometric graph and G' is required to be a 1-plane graph extending the embedding of G .*

Our reductions presented in Section 5 (from planar 3-SAT) work similar to [4] in the sense that the truth assignment of a variable is encoded in one of two possible perfect matchings on a subset of vertices whose degree must be augmented. The proof in [4] is specific to the

geometric setting, and it is not trivial to generalize it to the other two settings. Moreover, c -connected gadgets for $c \in \{2, 3, 4\}$ required delicate new designs to ensure that the desired behavior in the reduction [4] go through, especially when G is a triangulation.

Minimum-Degree Augmentation. In a k -connected graph G on $n > k$ vertices, the minimum vertex degree $\delta(G)$ is at least k . All new hardness reductions in Table 1 hinge on increasing the degree of a set of special vertices from c to k , using the minimum number of new edges, while maintaining planarity. As a consequence, we also prove that the following *min-degree augmentation* problems are NP-complete:

► **Corollary 1.2.** *Given a planar graph G with $\delta(G) = c$ and an integer k , find a minimum set F of new edges such that $G' = (V, E \cup F)$ is planar and satisfies $\delta(G') \geq k$.*

Flip Distance to a 4-Connected Triangulation. A *flip* (a.k.a. *edge exchange*) in a combinatorial triangulation (i.e., edge-maximal planar graph) is an elementary operation that removes one edge and inserts a new edge, producing another triangulation. The *flip diameter* on n vertices is the minimum number of flips that can transform any combinatorial triangulation into any other. Current bounds on the flip diameter use a canonical form, which is a triangulation with a Hamiltonian cycle. Since 4-connected triangulations are Hamiltonian [52] (see also [38, 44, 49] for extensions to beyond-planar graphs), the flip distance to a Hamiltonian or a 4-connected triangulation was studied [14, 16]. Bose et al. [13] posed the following problems: Given a combinatorial triangulation T , find the minimum number of flips that can take it to a 4-connected (resp., Hamiltonian) triangulation. Our hardness reduction for the $3 \rightarrow 4$ connectivity augmentation problem can be modified to show that finding the flip distance to 4-connectedness is also NP-complete (Corollary 1.3).

A *simultaneous flip* in a combinatorial triangulation is an operation that performs one or more edge flips simultaneously (each of which is a valid flip, and they can be performed independently) [24]. Bose et al. [12] showed that every triangulation on $n \geq 6$ vertices can be transformed into a 4-connected triangulation with a single simultaneous flip; the number of flipped edges is at most $\lfloor (2n - 7)/3 \rfloor$ [16]. We show that it is NP-complete to find a minimum simultaneous flip to 4-connectivity (Corollary 1.3). By inserting all simultaneously flipped edges (without removing any edges), we obtain a 4-connected 1-planar graph. In fact, finding a minimum augmentation to a 4-connected 1-planar graph in a triangulation is equivalent to finding a minimum simultaneous flip to 4-connectivity [12, Section 3].

► **Corollary 1.3.** *Given a combinatorial triangulation T and an integer τ , it is NP-complete to decide whether there exists a sequence of at most τ flips that transforms T into a 4-connected triangulation. It is also NP-complete to decide whether there is a single simultaneous flip of cardinality at most τ that transforms T into a 4-connected triangulation.*

In Section 6, we give an EPTAS for the first of the two problems (Theorem 6.1). We also give some intuition for why our techniques do not directly apply to the second problem.

Further Related Work. While we aim for increasing the *vertex-connectivity* of a given graph, analogous problems can be considered for *edge-connectivity* [5, 17, 33, 43, 50, 54]. Some of our techniques might extend to edge-connectivity, however, such adaptations are not immediate and are beyond the scope of this paper.

We have measured the distance to planarity by the local crossing number. However, there are many other notions of beyond-planarity. For example, García et al. [25] studied the connectivity augmentation problem for *geometric biplane graphs*, which admit a straight-line

drawing in the plane and a 2-coloring of the edges such that no two edges of the same color cross (a.k.a. graphs with geometric thickness 2). They showed that every sufficiently large point set in the plane in general position admits a 5-connected geometric biplane graph, and the connectivity bound 5 cannot be improved. Furthermore, every PSLG (other than a wheel or a fan) can be augmented into a 4-connected geometric biplane graph, and the connectivity bound 4 cannot be improved.

2 Tradeoff between Connectivity and Local Crossing Number

As a warmup exercise, we start with the following basic question, which corresponds to augmenting the empty graph to a k -connected graph with local crossing number at most ℓ .

► **Question 2.1.** *Given $\ell \in \mathbb{N}$, what is the maximum connectivity of an ℓ -planar graph?*

► **Proposition 2.2.** *For every $\ell \in \mathbb{N}$, the connectivity of an ℓ -planar graph is $O(\sqrt{\ell})$. This bound is the best possible and can be attained by a geometric graph on any point set in general position. Every set of $n \geq k+1$ points in the plane (in general position) admits a k -connected $O(k^2)$ -planar geometric graph; and this bound is the best possible.*

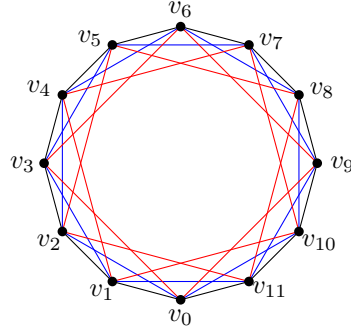
Proof. Let $G = (V, E)$ be a k -connected ℓ -planar graph on $n = |V|$ vertices. Then $\deg(v) \geq k$ for every $v \in V$, and so $|E| \geq kn/2$. Assume w.l.o.g. that $k \geq 8$, hence $|E| \geq 4n$. By the Crossing Lemma [3, 15, 40], the total number of crossings in any drawing of G is $\text{cr}(G) = \Omega(|E|^3/n^2) = \Omega(k^3n)$. By the pigeonhole principle, there is an edge with $\Omega(\text{cr}(G)/|E|) = \Omega(|E|^2/n^2) = \Omega(k^2)$ crossings; hence $\ell = \Omega(k^2)$ and $k = O(\sqrt{\ell})$.

For a matching lower bound, let V be the set of $n \geq k+1$ points in the plane in general position. Assume w.l.o.g. that they have distinct x -coordinates. Sort the points by x -coordinate and connect every point to its k neighbors on the left and right. The first $k+1$ points induce a complete graph, which is k -connected. By induction, the first i points also induce a k -connected graph for $k < i \leq n$. Any edge can only cross other edges with overlapping x -projections, so every edge has $O(k^2)$ crossings. ◀

Determining the precise dependency between k and ℓ remains largely open. One exception is the case $k = 1$: The maximum connectivity of a 1-planar graph is 7. Every 1-planar graph on $n \geq 3$ vertices has at most $4n - 8$ edges [45], so the minimum degree is at most 7, hence its connectivity is at most 7. This degree bound is tight: Biedl [9] constructed 1-planar graphs with minimum degree 7, which happen to be 7-connected. However, the connectivity of *optimal* 1-planar graphs (which have precisely $4n - 8$ edges) is either 4 or 6 [23, 48]; there are edge-maximal 1-planar graphs with $\frac{8}{3}n - O(1)$ edges, where the connectivity is only 2 [30].

Geometric graphs. We can determine the exact tradeoff between connectivity and local crossing number for geometric graphs when the vertices are in convex position. For integers $k, n \in \mathbb{N}$, $k < n/2$, the k -circulant graph $G = (V, E)$ is defined on the vertex set $V = \{v_0, \dots, v_{n-1}\}$, where $v_i v_j \in E$ if and only if the cyclic distance between i and j is at most k , that is, $\min\{|i - j|, n - |i - j|\} \leq k$; see Figure 2.

► **Proposition 2.3.** *For every $k \in \mathbb{N}$ and for $n \geq 2k$ points in convex position in the plane, the k -circulant graph is $2k$ -connected and is $(k^2 - k)$ -planar. This is the best possible: If a geometric graph on n points in convex position has minimum degree at least $2k$, then its local crossing number is at least $k^2 - k$.*



■ **Figure 2** The 3-circulant graph on $n = 12$ vertices.

Proof. Let $k \in \mathbb{N}$, and let $G = (V, E)$ be the k -circulant graph on n vertices, that is, the vertices are on a circle, and each vertex is adjacent to k neighbors in both cw and ccw direction along the circle. Elementary counting shows that every edge crosses at most $k^2 - k$ other edges. We show that G is $2k$ -connected. We need to show that $G - D$ is connected for any $D \subset V$ with $|D| = 2k - 1$. Let $s, t \in V \setminus D$. Then s and t decompose the circle into two arcs. One of them contains at most $k - 1$ vertices in D . We construct an st -path π_{st} by following this arc, and skip all vertices in D . Since we skip at most $k - 1$ consecutive vertices, all edges of π_{st} are present in $G - D$, and so $G - D$ is connected.

Now let $G = (V, E)$ be a geometric graph on $n \geq 2k$ points in convex position, with minimum degree at least $2k$. We define the *length* of an edge uv as the minimum number of edges in a path on the boundary of the convex hull. Then $|E| \geq kn$, and elementary counting shows that G has at least one edge of length at least k . Let e be a shortest edge among all edges of length at least k . We claim that if e has length $\ell \geq k$, then it has at least $k^2 - 3k + 2\ell \geq 2k^2 - k$ crossings. Assume w.l.o.g. that $V = \{v_0, \dots, v_{n-1}\}$ in cyclic order along the convex hull of V , and $e = (0, \ell)$. Every edge that crosses e has on endpoint in $\{v_1, \dots, v_{\ell-1}\}$ and one in $\{v_{\ell+1}, \dots, v_{n-1}\}$. Then $\sum_{i=1}^{\ell-1} \deg(v_i) \geq 2k(\ell - 1)$. We subtract the number of vertex-edge pairs (i, e') such that $1 \leq i \leq \ell - 1$ and e' does not cross e . Every such edge e' has length less than k . By symmetry, it is enough to count pairs (i, e') where e' is a “left” edge, that is, $e' = \{i, j\}$ and $j < i$. The number of such edges for $i = 1, 2, \dots, \ell - 1$ is $1, 2, 3, \dots, k - 1, k - 1, \dots, k - 1$, which sums to $\binom{k}{2} + (\ell - k)(k - 1)$. So edge e crosses at least $2k(\ell - 1) - 2\left(\binom{k}{2} + (\ell - k)(k - 1)\right) = k^2 - 3k + 2\ell \geq k^2 - k$ edges. This is a lower bound for the local crossing number of G . ◀

3 Augmentation for Topological Graphs

Every edge-maximal plane graph is 3-connected, but it may have separating triangles. For augmentation to 4-connectivity, we need to go beyond planarity.

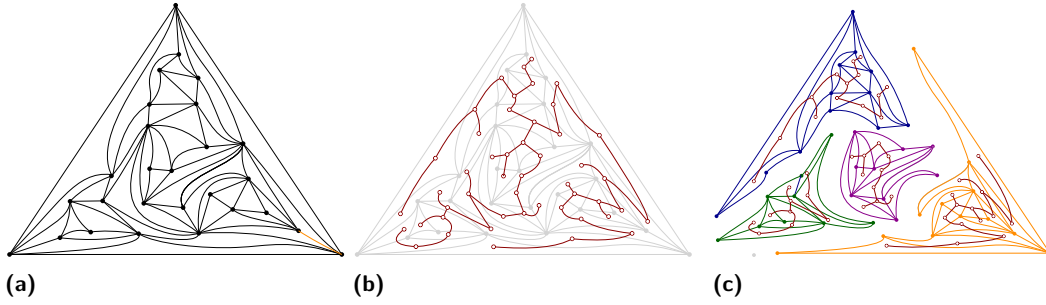
► **Proposition 3.1.** *Every plane graph G on $n \geq 6$ vertices can be augmented to a 4-connected 1-planar topological graph. If G is a triangulation, then $\lfloor (2n - 7)/3 \rfloor$ new edges suffice.*

Proof. Let $G = (V, E)$ be a plane graph on $n \geq 6$ vertices. We may assume that G is an edge-maximal plane graph (i.e., a triangulation). Cardinal et al. [16, Theorem 3] proved that G can be transformed into a 4-connected edge-maximal planar graph $G' = (V, E')$ using a *simultaneous flip* of at most $\lfloor (2n - 7)/3 \rfloor$ edges. Now $G = (V, E \cup E')$ is 4-connected (since it contains G') and 1-planar (an edge has a crossing iff it participates in a flip). ◀

For plane graph augmentation, we prove a tight tradeoff (matching Proposition 2.2).

► **Theorem 3.2.** *Every plane graph can be augmented to a k -connected $O(k^2)$ -planar topological graph, and this bound is the best possible.*

Proof. Without loss of generality assume we are given a plane triangulation $G = (V, E)$ and a $k \in \mathbb{N}$. The dual graph $\mathcal{D}$ of G is a planar graph of maximum degree 3. We can partition $\mathcal{D}$ into a collection $\mathcal{C} = \{C_1, \dots, C_t\}$ of connected subgraphs, each containing at least $2k - 1$ and at most $6k - 2$ nodes as follows. Consider an arbitrary spanning tree $\mathcal{T}$ of $\mathcal{D}$, and recursively partition $\mathcal{T}$ into subtrees by deleting an edge incident to a centroid node, until any further partition would produce a subtree with less of than $2k - 1$ nodes (see Figure 3).



■ **Figure 3** A triangulation on $n = 27$ vertices (a); a spanning tree $\mathcal{T}$ of the dual graph $\mathcal{D}$ (b); four clusters obtained for $k = 5$ (c).

For every $i = 1 \dots t$, let R_i denote the region formed by the union of triangles in C_i ; and let $V_i \subset V$ denote the set of all vertices of the triangles in C_i . By Euler's formula, applied for the triangles in C_i , we have $k \leq |V(C_i)| \leq 6k$. We call the vertex sets $V_1, \dots, V_t$ *clusters*. We also define a *cluster graph* $\mathcal{G}$, where the nodes correspond to the t clusters, and nodes V_1 and V_j are adjacent in $\mathcal{G}$ if C_i and C_j contain two triangles that share an edge. Note that a vertex $v \in V$ may be part of arbitrarily many clusters (e.g., if v is a high-degree vertex), yet each cluster is adjacent to $O(k)$ other clusters in $\mathcal{G}$.

We can augment G as follows. (1) Augment each cluster V_i to a clique, drawing all new edges along shortest paths in the region R_i . (2) For each pair of adjacent clusters, V_i and V_j , add a matching of size $k - |V_i \cap V_j|$ between $V_i \setminus V_j$ and $V_j \setminus V_i$, drawing the new edges along shortest paths in the region $R_i \cup R_j$. Let G' denote the augmented graph.

First, we show that G' is k -connected. Specifically, we show by induction on $i \geq 1$ that if i clusters induce a connected subgraph of the cluster graph, then the union of the clusters induce a k -connected graph in G' . The claim is clear in the base case $i = 1$. For the induction step, we use Menger's Theorem. Assume that the claim holds for $i - 1$ clusters. The union of i clusters is composed of two parts, A and B , where A is the union of $i - 1$ clusters, including cluster V_j , and $B = V_i$ is the i -th clusters. Both $G'[A]$ and $G'[B]$ are k -connected by the induction hypothesis, they share $|V_i \cap V_j| \geq 2$ vertices, and are connected by a matching of size $k - |V_i \cap V_j|$ (which is disjoint from the shared vertices). Let $a_1 = b_1, \dots, a_s = b_s$ be the shared vertices, and let $a_t b_t$, $t = s + 1, \dots, k$, be the matching with $a_i \in A \setminus B$ and $b_i \in B \setminus A$. Let $a \in A$ and $b \in B$ be arbitrary vertices. By Menger's Theorem, there are k internally disjoint paths from a to $a_1, \dots, a_k$ in $G'[A]$, hence k internally disjoint path from a to $b_1, \dots, b_k$ in $G'[A \cup B]$. Since $G'[B]$ is a clique, we obtain k internally disjoint paths between a and b in $G'[A \cup B]$. By Menger's Theorem, $G'[A \cup B]$ is k -connected.

Second, we claim that G' is $O(k^2)$ -planar. Consider an edge e in a cluster C . In the complete graph of C , edge e is crossed by $O(k^2)$ other edges. Cluster C is adjacent to $O(k)$ other clusters, each of which is connected to C by a matching of size $O(k)$. Edge e crosses

$O(k^2)$ edges in these matchings. Therefore, any edge in a cluster crosses $O(k^2)$ other edges in G' . Now consider an edge f of a matching between two adjacent clusters A and B . Then f can cross any of the $2 \cdot O(k^2)$ edges in the two cliques induced by clusters A and B . It can also cross the $2 \cdot O(k^2)$ edges in the $O(k)$ matchings between A (resp., B) and their adjacent clusters. Overall, any edge in G' has $O(k^2)$ crossings. $\blacktriangleleft$

4 1 $\rightarrow$ 3 Connectivity Augmentation for Plane Trees

In this section we present a linear-time algorithm for the following problem: Given an n -vertex plane tree T (i.e., with a fixed embedding), compute a 3-connected plane graph G with the minimum number of edges such that T is a spanning subgraph of G . That is, we solve the 3-connectivity augmentation of plane trees in the topological setting in linear time. We remark that Dhanalakshmi, Sadagopan, and Manogna [18] solve the problem for a tree T , where G is not necessarily planar. However, we could not fully verify the correctness of their algorithm. In their last step, the algorithm seems to introduce a matching on a long path of degree-2 vertices in T , which does not result in a 3-connected graph.

Let us start with a simple lower bound (which also appears in [18]) on the number of inserted edges. For a graph $G = (V, E)$ and integer $i \geq 0$, let $n_i(G) = |\{v \in V : \deg(v) = i\}|$ denote the number of vertices in G of degree i .

► **Observation 4.1.** *Let T be a spanning tree in a 3-connected graph G . Then G has minimum degree $\delta(G) \geq 3$. In particular, $|E(G)| - |E(T)| \geq \left\lceil \frac{2n_1(T) + n_2(T)}{2} \right\rceil$.*

We present an algorithm that meets the lower bound of Observation 4.1.

► **Theorem 4.2.** *Let T be an n -vertex plane tree with $n \geq 4$ and $n_i(T)$ vertices of degree i . Then there exists a plane n -vertex graph G with $T \subseteq G$ and $|E(G)| - |E(T)| = \left\lceil \frac{2n_1(T) + n_2(T)}{2} \right\rceil$.*

Moreover, such a G can be computed in $O(n)$ time.

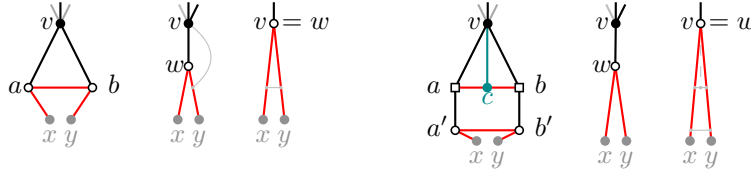
Proof. We present our construction of G as an inductive proof on n , which can be easily turned into a linear-time recursive algorithm. For simplicity, we call the edges of $E(G) \setminus E(T)$ the *new* edges and illustrate them in red. For the induction base of $n = 4$, we set $G = K_4$. So let us henceforth assume that $n \geq 5$. We use a stronger induction hypothesis, where we require that $n_2(T)$ is even; we later show how to handle the odd case. This assumption ensures that each leaf of T is incident to exactly two new edges, each vertex of degree 2 to exactly one new edge, and all other vertices to no new edges. We distinguish four cases.

Case 1: A vertex v has two consecutive neighbors a, b of $\deg(a) = \deg(b) = 1$.

Say a comes clockwise after b at v . We remove a and b from T . If v is not a leaf after this removal, then we add another leaf w at v ; otherwise v takes the role of w . We denote the new tree by T' and note that $n_2(T') = n_2(T)$, i.e., the number of degree-2 vertices remains the same. In order to apply the induction, we must also ensure that T' has at least 4 vertices if no new leaf was added, i.e., T has at least 6 vertices. This is the case as otherwise T corresponds to the tree T_5^* depicted in Figure 5, for which $n_2(T_5^*)$ is odd. In the obtained solution G' with $T' \subseteq G'$, vertex w is incident to two edges in $F' = E(G') - E(T')$, say wx comes clockwise after wy at w . Then we remove wy and insert ab and by .

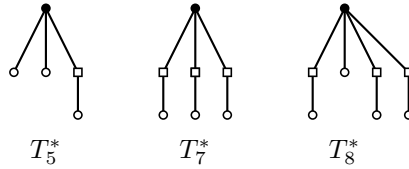
We define a *leg* as a vertex v in T of $\deg(v) = 2$ with neighbor a of $\deg(a) = 1$.

Case 2: A vertex v has two neighbors a, b that are legs which are either consecutive or have a leaf c between them.



■ **Figure 4** Illustration of Case 1 and 2 in the proof of Theorem 4.2.

We may assume that a comes clockwise after b , and if it exists, c comes after b and before a at v . Then we remove a, b , their adjacent leaves a', b' and c . If v is not a leaf after this removal, then we add another leaf w at v ; this ensures that v does not have degree 2. As before, we aim to recurse on the resulting tree T' . Note that $n_2(T') = n_2(T) - 2$ because a and b are removed and v has degree 1 or at least 3 in T' . We also ensure that T' has at least 4 vertices. In the case that we introduce a new leaf w , this is due to the fact that v has at least two more neighbors. In the other case, we must exclude the case that T' is a path on three vertices. However, this would imply that T is either the tree T_7^* or T_8^* , illustrated in Figure 5, both of which have an odd number of degree-2 vertices. Thus, we may apply induction on T' .

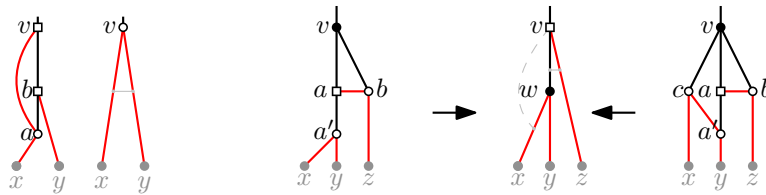


■ **Figure 5** Illustration for the (non-existing) base cases in the proof of Theorem 4.2.

In the obtained solution G' , vertex w has two incident edges in $F' = E(G') - E(T')$, say wx comes clockwise after wy at v . Then we remove wx, wy and insert $a'x, a'b'$, and $b'y$. If c exists in T , we insert ac and cb , otherwise we insert only ab .

Case 3: A vertex v with $\deg(v) = 2$ has a neighbor b that is a leg adjacent to leaf a .

In this case we delete a and b and recurse on $T' = T - \{a, b\}$; note that $n_2(T') = n_2(T) - 2$. Moreover, if T' had only 3 vertices, then T would correspond to a path on 5 vertices for which $n_2(T)$ would be odd. Hence, we may apply the induction. In the obtained solution G' , vertex v has two incident edges in $F' = E(G') - E(T')$, say vx and vy . Then we remove vx, vy and insert va, ax , and by .



■ **Figure 6** Illustration of Case 3 and 4 in the proof of Theorem 4.2.

Case 4: A vertex v (with $\deg(v) = 3$ or 4) has a neighbor p with $\deg(p) \geq 3$, one neighbor a which is a leg and otherwise only leaves.

By Case 1, we may assume that no two leaves are consecutive. Hence, either v has exactly one leaf b or two leaves b, c which are separated by p and a , see also Figure 6. In this case we delete a, b, c and the leaf a' of a , and insert a new leaf w at v . We recurse on the resulting tree T' . Note that $n_2(T') = n_2(T)$ as we deleted a and v has degree 2 in T' . Moreover, due to the fact that p has $\deg(p) = 3$, T' has at least 4 vertices. In the solution G' obtained by induction, vertex v is incident to one edge, say vz , and vertex w to two edges, say wx and wy , in $F' = E(G') - E(T')$. Then we remove wx, wy, wz and insert $a'y, ab$, and bz . If c does not exist, we insert $a'x$ and otherwise $a'c, cx$.

This completes the construction of G with $T \subseteq G$. To show that the case distinction is exhaustive, root the tree T at an arbitrary vertex and consider a lowest vertex v that is neither a leaf nor a leg. All but one neighbor p of v , are either leaves or legs. If v has $\deg(v) = 2$, then Case 3 applies. Otherwise, apart from p , v has two consecutive leaves (Case 1), two legs which are consecutive or have a leaf between them (Case 2), or one leg and otherwise only leaves (Case 4).

It remains to argue that the resulting graph G is 3-connected. This is clear in the base case $G = K_4$. In each remaining case, it suffices to observe that G is obtained from a 3-connected graph G' by a local modification that is a sequence of so-called *BG-operations*; these operations were introduced by Barnette and Grünbaum [8] and preserve 3-connectivity. In a $(1, 2)$ -operation we subdivide an edge xy by a new vertex w and insert an edge vw with $v \neq x, y$. In a $(2, 3)$ -operation we subdivide edge xy by vertex a and edge vw by vertex b and insert the edge ab . Figures 4 and 6 illustrate how up to three of these operations suffice.

Lastly, we show how to handle the case that $n_2(T)$ is odd. We choose an arbitrary vertex v of degree 2 and contract one of its incident edges to obtain a tree T' . Using our above machinery, we obtain a 3-connected graph G' with $|E(G')| - |E(T')| = \left\lceil \frac{2n_1(T) + n_2(T) - 1}{2} \right\rceil = \left\lceil \frac{2n_1(T) + n_2(T)}{2} \right\rceil - 1$. Now we reintroduce v by a $(1, 2)$ -operation on its edge, connecting v to an arbitrary non-neighbor in an incident face. As this preserves 3-connectivity, this completes the proof. $\blacktriangleleft$

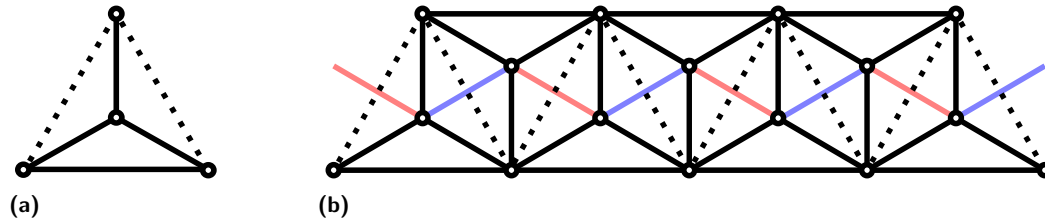
5 Hardness Results

In this section we prove our main hardness results (Theorem 1.1 and Corollary 1.3). We reduce from planar 3-SAT. A 3-SAT instance is given by a boolean 3-CNF formula φ , with n variables $x_1, x_2, \dots, x_n$ and m clauses $C_1, C_2, \dots, C_m$. In planar 3-SAT, φ can be represented as a planar bipartite graph with vertices for every variable and clause, and an edge (x_i, C_j) if x_i or $\neg x_i$ is a literal in clause C_j . This is a well studied SAT variant proven NP-complete by Lichtenstein [41]. We have three reductions that respectively cover the $2 \rightarrow 3$, $3 \rightarrow 4$ and $4 \rightarrow 5$ augmentations. Note that if the $c \rightarrow k$ augmentation problem is NP-complete, so is the $c' \rightarrow k$ problem for $c' < c$ because every c -connected graph is also c' -connected. All three reductions are similar: constructed from three gadgets (variable, clause, and literal) that are put together in the standard way. Due to the similarities between reductions, we will address only one of them here and relegate the other two to the full version of this paper. In particular, we only provide our reduction for $3 \rightarrow 4$ augmentation as it is the best representative of the three reductions.

5.1 $3 \rightarrow 4$ Connectivity Augmentation is NP-Complete

Our reduction works for all settings (abstract, topological and geometric) since 3-connected graphs have a unique combinatorial embedding and the reduction can be embedded geometrically. We use three gadgets, a variable, a literal, and a clause gadget. We ensure that

the entire construction is 4-connected except for some select degree-3 vertices. The variable gadget is shown in Figure 7b. The “building block” of our construction is essentially a wheel graph on four vertices W_4 . We modify this W_4 , removing two edges so that one vertex is degree-3, one is degree-1, and two are degree-2 (see Figure 7a). We call the degree three vertex the *central* vertex and the others the *boundary* vertices. We construct a variable gadget by “gluing” an even number of these modified W_4 ’s together so any two consecutive W_4 have two shared boundary vertices. Hence every boundary vertex is identified with two separate W_4 ’s, while the central vertices are unique to each W_4 , so every central vertex is precisely degree three and must therefore be augmented.



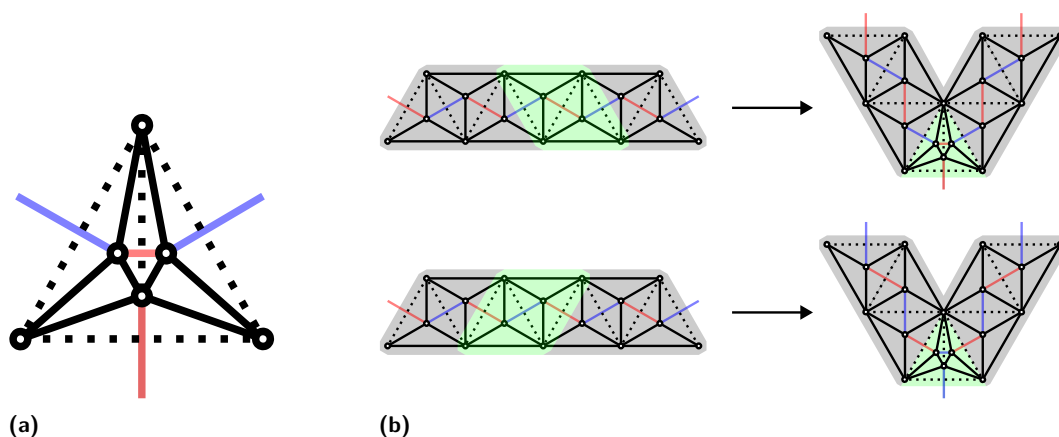
■ **Figure 7** (a) Modified W_4 , removed edges indicated with dashed line. (b) A variable gadget. Red segments indicate a false assignment, blue indicate a true assignment.

► **Lemma 5.1.** *The variable gadget (Figure 7b) constructed with $2k$ modified W_4 ’s can be minimally augmented with k edges, and there are two such augmentations.*

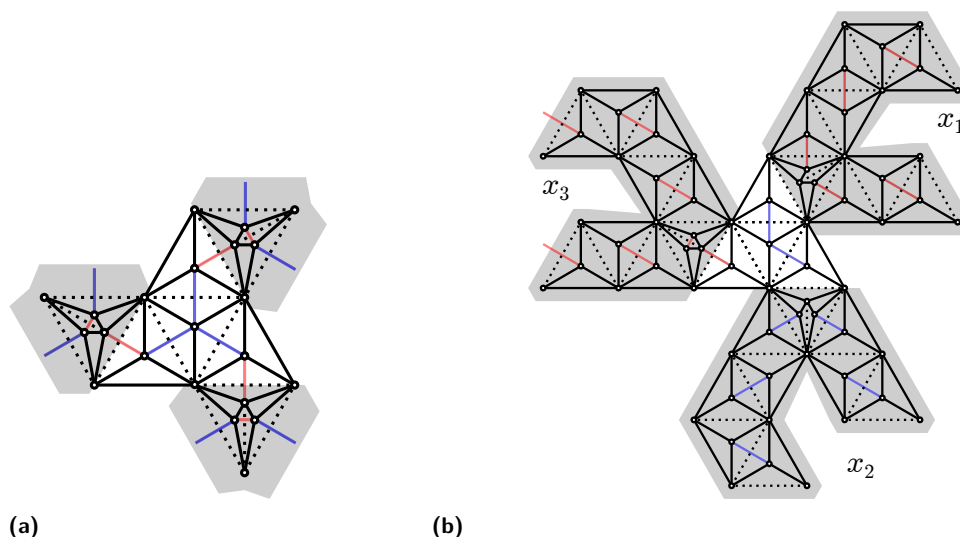
Proof. As each central vertex is still unique to its W_4 , the boundary vertices of each W_4 form a 3-cut disconnecting their central vertex. Therefore, augmenting this gadget to be 4-connected requires adding an edge from each central vertex to a vertex in a separate W_4 . This augmentation must be planar and so each central vertex can only be connected to the central vertices of its two adjacent W_4 ’s. Note, once a central vertex is connected to another central vertex, each of their adjacent boundary vertices no longer form a 3-cut. So we only need to connect every other pair of central vertices, and in fact this would be the minimal augmentation for this gadget. Each central vertex must be connected to at least one other vertex, and they can only be connected to other central vertices. Therefore connecting each central vertex to precisely one central vertex must be minimal, and clearly the number of edges required is precisely half the number of central vertices.

Note there are two unique ways to do this. Arbitrarily identify one W_4 as the “initial” and orient the variable cycle from this W_4 . The two possible augmentations are therefore either each central vertex is attached to its clockwise neighbor, or each central vertex is adjacent to its counter clockwise neighbor. If clockwise neighbors are adjacent we say the variable is set to “true” and we say it is “false” if counter clockwise neighbors are connected. ◀

A literal gadget (Figure 8a) is composed of six vertices, and similar to the modified W_4 ’s used in the variable gadget. Again, as with the variable gadget, the three boundary vertices form a 3-cut that disconnects the three central vertices. As depicted in Figure 8b literal gadgets are “inserted” into a variable gadget, essentially replacing two of W_4 ’s of a variable. Consider a literal gadget inserted into a variable x_i . There are only two possible locally minimal edge augmentations of the literal gadget, as depicted in Figure 8b. For a positive literal (Figure 8b(top)) if x_i is true (resp., false) we connect the degree-3 vertices with the blue (resp., red) matching.



■ **Figure 8** (a) A positive literal gadget (b) How a literal gadget is “inserted” into a variable gadget. Two W_4 's of the variable are replaced by one literal. A clause gadget is adjacent to the exposed dotted line. On the top is a positive literal, the bottom shows a negated literal.



■ **Figure 9** (a) A clause gadget (b) Example where two variables are set to true and one is set to false. Variables are highlighted in grey.

The clause gadget (Figure 9) requires 3 new edges, and is achievable iff a literal is true.

► **Lemma 5.2.** *The clause gadget shown in Figure 9 can be augmented with exactly three edges if and only if one adjacent literal gadget is true. Otherwise 4 edges are necessary.*

Proof. Let c be a clause of φ on three variables x_i , x_j , and x_k . For each literal of c we “glue” a term K_4 to the corresponding variable gadget. We take two adjacent W_4 's of the variable gadget and replace them with a literal gadget. (Figure 8b).

To augment the clause gadget, for each of its W_4 's the central vertex must be connected to a vertex of another W_4 . Note, the central vertex of the value W_4 can only connect to one of the central vertices of the terms. While each term W_4 can either have its central vertex connected to a vertex in an adjacent literal gadget or to the central vertex of the value W_4 .

Therefore, augmenting this gadget requires at least three edges, one edge for each term. We now show that this can be augmented exactly three edges only if at least one adjacent literal gadget is true.

If a term is adjacent to a false literal then the only possible minimal augmentation is to connect the central vertex of the term to a central vertex of the literal gadget. This would augment both the term and the literal by adding only one edge. Any other augmentation would require at least two edges, we would have to connect a central vertex of the literal to a variable W_4 and we would have to connect the central vertex of the term to the central vertex of the value of the clause.

Note that a true literal gadget is already 4-connected. Therefore, if a term W_4 is adjacent to a true literal then we can connect its central vertex to the central vertex of the value of the clause, augmenting both to be 4-connected.

Therefore we can augment a clause gadget with only 3 edges if and only if it is adjacent to a true literal gadget. Otherwise we require 4 edges. If every term is adjacent to a false literal, then each of the false literal gadgets require one edge and the value of the clause requires an edge. As the augmentation must be planar, these edges are disjoint, and so we need to add four edges. So, c can be augmented with three edges if and only one adjacent literal gadget is true. ◀

With these three gadgets we can reduce from planar 3-SAT, giving the following lemma.

► **Lemma 5.3.** *Given a 3-connected planar graph $G = (V, E)$ and an integer τ , it is NP-hard to decide whether there is a set F of edges with $|F| \leq \tau$ such that $G' = (V, E \cup F)$ is a 4-connected planar graph.*

Proof. Recall that we consider the decision version of this problem. I.e. given a 3-connected plane graph G , can at most τ edges be added to form a 4-connected plane graph G' ?

Given a planar 3-SAT formula φ with n variables and m clauses. Let P be a planar embedding of φ . We now construct an instance of 3-4 graph augmentation, starting with a 3-connected plane graph.

Take every variable's embedding in P each will form a star graph. For each variable we create a corresponding variable gadget by taking a Eulerian circuit of each of these stars and replacing it with a variable gadget. These gadgets are cycles of an even number of modified W_4 's. Next replace every clause node with a clause gadget and for each of the three literals insert corresponding literal gadgets into the relevant variable gadgets, replacing two W_4 's. We can then compute what the correct value of k is using Lemmas 5.1 and 5.2. That is, k is half the number of W_4 's that occur in all variable gadgets plus 3 for each clause gadget ($3m$).

Now given a satisfying assignment S for φ we can augment each variable gadget accordingly and then if S is a satisfying assignment each clause gadget should have an adjacent true literal gadget. Therefore by Lemma 5.2 each clause can be augmented with 3 edges. So G can be augmented with at most τ edges.

If there is a set of τ edges, where τ is half the number of variable W_4 's plus 3 times the number of clauses that planarly augments G . Then there is a corresponding satisfying assignment of φ . By Lemma 5.1 there are only two ways to minimally augment each variable gadget, corresponding to false or true assignments respectively. Therefore, each variable gadget will uniquely encode a corresponding variable assignment, this gives some possible solution S' for φ . By Lemma 5.2 we will only be able to augment each clause gadget with three edges if one of its adjacent literal gadgets is true, otherwise at least one will require 4 edges. So we will only have an augmenting set of τ edges if every clause of φ has at least one true literal. So if there is an augmenting set of τ edges we can get a satisfying assignment S' for φ by checking each variable gadget.

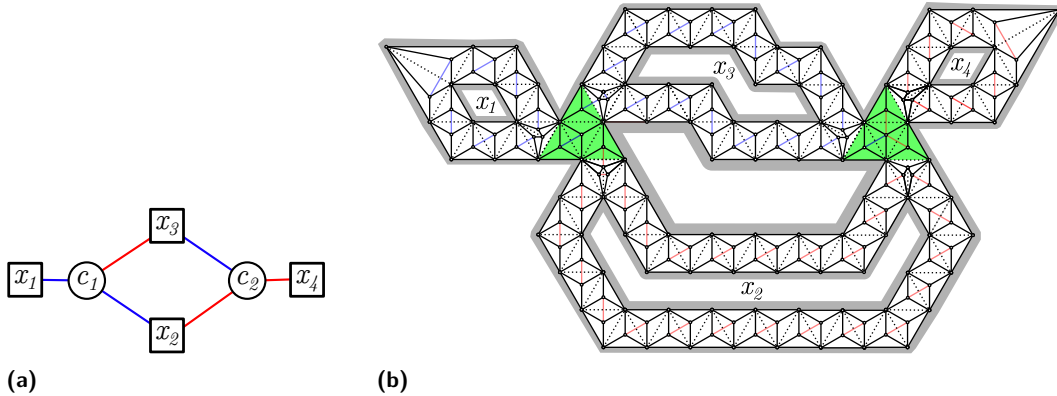
We conclude that φ has a satisfying assignment S if and only if G can be augmented with at most τ edges. $\blacktriangleleft$

► **Corollary 5.4.** *Lemma 5.3 still holds if G is a topological (or geometric) plane graph and if G' is required to be a compatible topological (resp., geometric) plane or 1-plane graph.*

Note, every argument that was made was independent of the magnitude or drawing of any edge (as long as they are planar). Hence our reduction easily extends to the topological and geometric settings, by just embedding each gadget (and therefore G) as depicted. The reduction is also easily adapted to the 1-planar setting. In each of our gadgets, we included dashed lines, like the ones between every W_4 in the variable gadget. By replacing these with an edge in G , augmenting G will necessarily result in a 1-planar graph.

► **Corollary 5.5.** *Given a combinatorial triangulation T and an integer τ . Determining if there is a series of $\leq \tau$ flips that transforms T into a 4-connected triangulation T' is NP-complete.*

This is a simple extension of our main reduction. Note, every black dotted line corresponds in our gadgets (Figure 7, Figure 8, and Figure 9) were an edge of G . Then G is a combinatorial triangulation, and each augmenting edge can be realized by a flip of one of these dotted edges. After adding these dotted lines as edges of G , our arguments from the $3 \rightarrow 4$ augmentation reduction easily translate.



■ **Figure 10** (a) A planar 3-SAT formula $\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_4)$ (b) The graph G constructed from φ during the reduction, x_1 and x_3 are assigned true, and x_2 and x_4 are set to false. Clause gadgets are highlighted in green for legibility.

6 EPTAS for Flip Distance of Triangulations to 4-Connectivity

In this section, we provide an EPTAS for making plane triangulations 4-connected with a sequence of edge flips.

► **Theorem 6.1.** *For every $\epsilon > 0$, there is an $n \cdot 2^{O(1/\epsilon)}$ -time algorithm that, for a given triangulation G on $n \geq 6$ vertices, returns a sequence of edge flips that turns G into a 4-connected triangulation, and the length of the sequence is at most $1 + \epsilon$ times larger than necessary.*

We first reduce the problem to a certain hitting set problem as suggested by Bose et al. [13]. The reduction relies on the following.

► **Lemma 6.2** (Mori et al. [42]). *Let G be a plane triangulation on $n \geq 6$ vertices, T a separating triangle of G and e an edge of T . Then flipping e destroys the separating triangle T and does not create any new separating triangles in G , provided e is incident to multiple separating triangles or none of the three edges of T are incident to multiple separating triangles.*

This allows one to easily reduce the problem of finding an optimal sequence of flips to a certain hitting set problem on the edges of the triangulation.

► **Lemma 6.3.** *Let G be a plane triangulation on $n \geq 6$ vertices. Let E' be a set of edges of G such that every separating triangle in G is incident to at least one edge in E' , and let τ be the minimum size of any such set.*

Making G 4-connected requires at least τ edge flips. Moreover, given E' , one can compute a sequence of edge flips of length at most $|E'|$ making G 4-connected in $O(n)$ time.

Proof. If F is a sequence of edge flips making G 4-connected, then every separating triangle in G must be incident to at least one edge in E' (otherwise the missing separating triangle remains in G after all flips have been performed, and the graph is not 4-connected). Thus, F has length at least τ .

Now consider the edges of E' in some arbitrary order $e_1, e_2, \dots, e_{|E'|}$. Define a sequence of edge flips as follows. Let $1 \leq i \leq |E'|$, and assume the first $i - 1$ edges of E' have already been processed. Then for the e_i , do the following:

- if e_i is not present in G (because it has already been flipped previously) or does not bound any separating triangles in G , don't flip any edge;
- otherwise, if e_i is incident to multiple separating triangles in G , flip e_i ;
- otherwise, if none of the 3 edges of the unique separating triangle T incident to e are incident to any other separating triangle, flip e_i ;
- otherwise, flip an edge of T incident to at least 2 separating triangles.

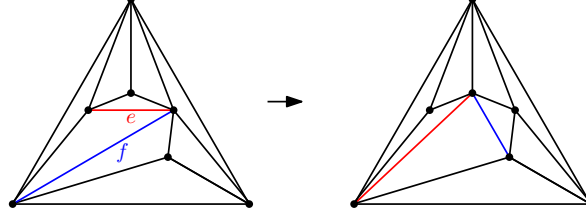
By the definition of this sequence and of E , it is clear that for any separating triangle in G , at least one of its 3 edges gets flipped. Moreover, every edge that is flipped obeys the conditions of Lemma 6.2 (at the time it is flipped), and thus, no new separating triangle is introduced and the resulting graph is 4-connected.

Let us describe how to carry out the procedure in linear time. For each edge, we will maintain a set of pointers to the separating triangles it bounds in a doubly-linked list. Each separating triangle also has corresponding back-pointers, so that from a separating triangle we can access the three pointers pointing to it in constant time. We can initialize this structure by computing the set of separating triangles incident to each edge in $O(n)$ total time (this can be done for example by computing a 4-block decomposition of the triangulation in linear time [36]).

For a given edge, we can then test in constant time if it bounds 0, 1 or at least two separating triangles. In the case it bounds exactly one separating triangle T , we can test in constant time whether this is also the case for the two other edges bounding T . Each time we flip an edge e , we go through the set of separating triangles it used to bound by traversing its associated doubly-linked list and following the pointers it stores. For each such separating triangle, we delete it from the doubly-linked lists of the three edges which (used to) bound it by following the back-pointers. This costs constant time per deleted separating triangle.

Because no new separating triangle is created at any point, every separating triangle is deleted once and the triangulation starts out with $O(n)$ separating triangles, the total runtime is $O(n)$. ◀

It is tempting to try to carry out the reduction by simply flipping the edges of E' in some arbitrary order (or perhaps skipping an edge if it is no longer incident to any separating triangle by the time it is considered). However, this strategy may fail as illustrated in Figure 11. Thus, a more careful strategy is needed. Our algorithm flips only edges that bound separating triangles and do not create any separating triangle, thus strictly decreasing the number of separating triangles with each flip.



■ **Figure 11** While $\{e, f\}$ is a minimum set of edges hitting every separating triangle, flipping e then f fails to make the triangulation 4-connected, as a new separating triangle is introduced.

► **Lemma 6.4.** *Given a plane graph G of treewidth $\leq k$ with n vertices, and a set C of 3-cycles of G , there is a $n \cdot 2^{O(k)}$ -time algorithm that computes a minimum cardinality edge set E' such that every 3-cycle in C is incident to at least one edge in E' .*

Proof. Recall that a tree decomposition of G is a mapping of G into a tree T_G whose vertices we shall call *bags*. A bag B is associated with a subset of vertices V_B . Every vertex appears in at least one bag and induces a single subtree in T_G , and if two vertices are adjacent in G then there is a bag in T_G containing both vertices. The width of T_G is the size of its largest bag. We root T_G arbitrarily. We can build a tree decomposition of G of width $O(k)$ in $n \cdot 2^{O(k)}$ time with the following properties [11, 39]:

- the number of bags is $O(n)$;
- the maximum degree of a bag is 3;
- for a bag with two children, all three bags have identical subsets of vertices; and
- for a bag with a single child the two subsets differ by one.

Since G is planar, each bag defines an induced subgraph with $O(k)$ edges. Note that, by definition, if G has a clique, these vertices must appear in a bag of T_G . Thus, every 3-cycle in C must appear in at least one bag of T_G . We compute E' using dynamic programming as follows. We say that a subset S of edges *satisfies* a subtree of T_G if every 3-cycle of C that appears in a bag of the subtree contains at least one edge in S . For each bag B and for each subset S of the $O(k)$ edges in V_B , we define a subproblem $A_{B,S}$ of computing the minimum cardinality set S' of edges satisfying the subtree rooted at B subject to choosing the edges in S in the induced subgraph of V_B . Thus there are $n \cdot 2^{O(k)}$ total subproblems. By our degree-3 assumption, the children of parents of high-degree simply receive the information from their parent without making any choices. The child of a low degree bag has to make choices about the incident edges to the potential new vertex v . The degree of v in the induced subgraph is $O(k)$ which implies that we have to make $2^{O(k)}$ binary choices. Some of these choices are fixed: if the addition of v closes a 3-cycle in C , the cycle has two incident edges to v and the choices are valid only if at least one edge of the separating triangle is chosen. Therefore to compute each subproblem there is $k \cdot 2^{O(k)} = 2^{O(k)}$ nonrecursive work. Since the tree decomposition has $O(n)$ nodes, this approach leads to a $n \cdot 2^{O(k)}$ runtime. ◀

We can now give the proof of Theorem 6.1.

Proof of Theorem 6.1. The idea is to apply Baker’s layer shifting technique [7] to the hitting set formulation of the problem. We actually solve the more general problem of hitting an arbitrary subset C of 3-cycles of G , given as an input.

Assume without loss of generality that $\epsilon \leq 1/2$ and set $k = \lceil 1/\epsilon \rceil$. Choose an arbitrary vertex v_0 of G and label all vertices of G according to their (shortest path) distance to v_0 . We call the set of vertices at a specific distance from v_0 a layer of G . For $0 \leq i < k$, assign the color i to all edges incident to two vertices labeled $(i \bmod k)$ (edges joining two vertices in consecutive layers do not get assigned a color).

Let E^* be a smallest subset of edges such that every 3-cycle in C is incident to at least one edge in E^* . Notice that there is at least one of the k colors, say, c , for which at most $|E^*|/k$ edges from E^* get assigned the color c .

Now imagine decomposing G into the subgraphs $G_0, \dots, G_p$ each induced by $k+1$ consecutive layers, starting at a layer whose label is $(c \bmod k)$ (except eventually for the first subgraph which starts at layer 0, and the last subgraph which might consist of fewer than $k+1$ consecutive levels). Note that consecutive subgraphs overlap at layers $(c \bmod k)$. Every 3-cycle in C appears in one of the subgraphs, as it necessarily has its 3 vertices in at most 2 consecutive layers, and every pair of consecutive layers appears in one of the subgraphs. Taking the union of optimal solutions for the problem on each subgraph (with C restricted to the 3-cycles which appear in that subgraph) thus gives a valid solution for G . Let $E_1, \dots, E_p$ be such optimal solutions, and for $1 \leq j \leq p$ let $E^*[G_j]$ denote the subset of edges in E^* which appear in G_j . We have that

$$\begin{aligned} \left| \bigcup_{1 \leq j \leq p} E_j \right| &\leq \sum_{1 \leq j \leq p} |E_j| \leq \sum_{1 \leq j \leq p} |E^*[G_j]| \\ &\leq |E^*| + |E^*|/k \leq (1 + \epsilon)|E^*|, \end{aligned}$$

where the second inequality stems from the fact that $E^*[G_j]$ is a valid solution for G_j , the third from the fact that only the edges of color c get counted twice (and there are at most $|E^*|/k$ such edges) and the last from the choice of $k = \lceil 1/\epsilon \rceil$.

The graphs $G_0, \dots, G_p$ are all $(k+1)$ -outerplanar and thus have treewidth $O(k)$ [10]. Using Lemma 6.4, we can thus solve the problem optimally for each G_j in $|V(G_j)|2^{O(1/\epsilon)}$ time. The total runtime is then $\sum_{1 \leq j \leq p} |V(G_j)|2^{O(1/\epsilon)} = n \cdot 2^{O(1/\epsilon)}$. We can carry out the described procedure for each value of $1 \leq c \leq k$ (note that we don’t know the right value of c to choose *a priori*) and take the best solution. This adds a factor of $1/\epsilon$ that gets absorbed by the exponential. The previous discussion, together with Lemma 6.3, then gives the sought result. $\blacktriangleleft$

While Baker’s method of decomposing a planar graph into overlapping layers of small treewidth and combining the solutions is classical, it is interesting to note that a direct adaptation fails for the problem of approximating a smallest set of edges which make a triangulation 4-connected when flipped simultaneously (or the equivalent problem of augmenting a triangulation to 1-planar 4-connected via edge insertions). This is because, while this problem is still efficiently solvable for small treewidth, taking the union of solutions might lead to selecting edges which are not simultaneously flippable. Perhaps this could be overcome by combining the solutions more carefully, but we leave the problem of finding a PTAS for the simultaneous case open here.

Previous results. Accornero, Ancona and Varini [2] present an algorithm and claim it computes the minimum number of edges hitting all separating triangles in a given triangulation. Note that this result together with our reduction in Section 5 (Corollary 1.3) would imply

$P=NP$. Although their algorithm computes a set of edges that hits all separating cycles, we claim this set is not minimum. We attempt here to give a short intuition for that. In short, their algorithm computes (the equivalent of) a 4-block tree of the input triangulation T , a decomposition of T into maximal 4-connected components (4-blocks) hierarchically organized as the nesting structure of separating triangles that bound the maximal components. At each level, the edges are weighted with the cost of the optimal solution for the children nodes that contain said edge in their outer face. Note that every edge bounds at most two separating triangles in the next level of the tree and, therefore, there are at most two children associated with the same edge. At the root of the tree (and recursively in every node) they then compute the minimum weight subset of edges that hit every separating triangle in the maximal 4-block associated with the node. In [2, Theorem 3] there is an implicit assumption that the optimal solution contains a single edge of each separating triangle, since the weights are computed based on the cost of recursive solutions that flip a specific edge. That is not true, for example, in our reduction where the optimal solution may flip two edges of a separating triangle bounding the literal gadget (blue edges in Figure 8).

7 2 $\rightarrow$ 3 Augmentation for Geometric Triangulations of n Points in Convex Position

Given a triangulation $G = (V, E)$ of n points in convex position and a positive integer ℓ , we can compute the *minimum* number of edges that augment G to a 3- or 4-connected ℓ -planar graph (or report that there is no such augmentation) by dynamic programming.

The key observation is a simple combinatorial characterization of 3- or 4-connected augmentation. We introduce some terminology. An edge $e \in E$ is a *diagonal* if it is not an edge of the convex hull of V ; and $e \in E$ is an *ear* if there is exactly one point in V in an open halfplane bounded by the line spanned by e .

► **Proposition 7.1.** *Let $G = (V, E)$ be a straight-line triangulation on n points in convex position, and let $G' = (V, E \cup F)$ be a geometric graph. Then*

- *G' is 3-connected if and only if every diagonal in E crosses an edge in F ;*
- *G' is 4-connected if and only if every diagonal in E crosses at least two edges in F ; and every diagonal in E is either an ear or crosses at least two disjoint edges in F .*

Proof. If G' is 3-connected, then the two endpoints of a diagonal $e \in E$ cannot be a 2-cut, and so some edge $f \in F$ connects the two components of $G - ab$. Since the vertices are in convex position, then e and f cross.

Conversely, suppose that every diagonal in E crosses an edge in F , and G' has a 2-cut $\{a, b\}$. Then $\{a, b\}$ is already a 2-cut in G , so ab is a diagonal, and $G - ab$ has exactly two connected components. However, an edge $f \in F$ that crosses ab connects the two components, contradicting the assumption that $\{a, b\}$ is a 2-cut in G' .

If G' is 4-connected, then the two endpoints of a diagonal $ab \in E$ and an endpoint of a crossing edge $cd \in F$ cannot form a 3-cut. This leaves us with two possibilities: either c or d is the only vertex in one of the components of $G - ab$ (in this case ab is an ear), or there an edge in $f_1 \in F$ that crosses ab and not incident to c , and an edge $f_2 \in F$ that crosses ab and not incident to d . Now if $f_1 = f_2$, then cd and $f_1 = f_2$ are disjoint, otherwise f_1 and f_2 are disjoint. In both cases, two disjoint edges in F cross ab .

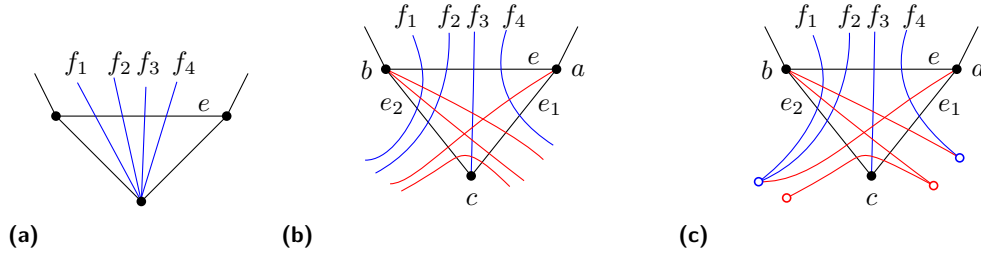
Conversely, suppose that G' satisfies the conditions above but has a 3-cut $\{a, b, c\}$. If none of ab , ac and bc is diagonal in E , then $G - \{a, b, c\}$ is connected, so $\{a, b, c\}$ would not be a 3-cut. First, assume that ab , ac , and bc are all diagonal in G . Then $G - \{a, b, c\}$ has exactly

three components. However each edge of $\Delta(a, b, c)$ crosses two edges in F , and in particular at least one edge that is not incident to $\{a, b, c\}$. Thus, each of the three components of $G - \{a, b, c\}$ is connected to another component in G' , and so $G' - \{a, b, c\}$ is connected.

Next, assume that only one edge of $\Delta(a, b, c)$, say ab , is a diagonal of E . Then $G - \{a, b, c\}$ has exactly two components, separated by the line spanned by ab . In particular, if ab is an ear, then c cannot be the only vertex on one side of the line spanned by ab . In this case, ab crosses at least two edges $f_1, f_2 \in F$. At most one of them is incident to c : This is clear if ab is not an ear, and f_1, f_2 are disjoint. If ab is an ear, then f_1, f_2 are incident to a single vertex on one side of ab , but this vertex is not c , and so one of f_1 and f_2 is not incident to c . Since F contains an edge between the two components of $G - \{a, b, c\}$, then G' is connected. ◀

► **Theorem 7.2.** *For any $k \in \{3, 4\}$ and any constant $\ell \in \mathbb{N}$, there is an algorithm that, given a PSLG triangulation $G = (V, E)$ on n points in convex position, can find a minimum set F of new edges in $O(n)$ time such that $G' = (V, E \cup F)$ is a k -connected ℓ -planar geometric graph, or reports that there is no such graph G' .*

Proof. Let e_0 be an arbitrary edge of the convex hull of V , and let E_0 be the subset of E comprising e_0 and all diagonals in E . We define a partial order $\preceq$ on E_0 : We say that $e \preceq f$ if both f and e_0 are in the same closed halfplane of bounded by the line spanned by e . (In particular, $e \preceq e_0$ holds for all $e \in E_0$. The poset $(E_0, \preceq)$ defines a tree, rooted at e_0 , which is binary tree rooted at e_0 . For a dynamic programming algorithm, we define the subproblems that correspond to the edge $e \in E_0$.



■ **Figure 12** Illustration for a dynamic programming algorithm: An ear diagonal e (a); a non-ear diagonal e for $k = 3$ (b), and for $k = 4$ (c).

3-connectivity. For an edge $e \in E_0$, integer $q \in \{0, \dots, \ell\}$ and a vector $\vec{c} \in \{1, \dots, \ell\}^q$, let $\text{OPT}_3(e, q, \vec{c})$ denote the minimum size of a set of new edges F such that $G' = (V, E \cup F)$ has the following properties:

- Edge e crosses exactly q edges in F : $f_1, \dots, f_q$;
- for $i = 1, \dots, q$, edge $f_i \in F$ crosses at most c_i edges $h \in E_0$, $h \preceq e$; and
- every edge $h \in E_0$, $h \preceq e$, crosses an edge in F (cf. Proposition 7.1 for $k = 3$).

If no such set F exists, then we set $\text{OPT}_3(e, q, \text{vecc}) = \infty$.

Our DP algorithm considers all edges $e \in E_0$ in increasing order in the poset $(E_0, \preceq)$. If e is a minimal element of the poset $(E_0, \preceq)$, then e is an ear, and $\text{OPT}_3(e, q, \vec{c}) = q$ for all $\vec{c}$; see Figure 12a. If e is not minimal, then we may assume that $e = ab$ in a triangle $\Delta(a, b, c)$, where $e_1 = ac$ or $e_2 = bc$ is also a diagonal in E_0 . To compute $\text{OPT}_3(e, q, \vec{c})$, we compare all possibilities in the triangle $\Delta(a, b, c)$ by brute force: Each of the q edges that cross e may cross e_1 or e_2 , or end at vertex c . Furthermore, any edge that crosses e_1 (resp., e_2) may end at b (resp., a) or cross e or e_2 (resp., e_1). For each combination, we determine whether they are feasible (i.e., satisfy all constraints), and let $\text{OPT}_3(e, p, \vec{c})$ be the minimum size of a

feasible solution; see Figure 12b. Importantly, the order in which e crosses $f_1, \dots, f_q$ is not specified – we count only unavoidable crossings: The edges in $\{f_1, \dots, f_q\}$ that are incident to c must cross all edges that cross both e_1 and e_2 , or cross e_1 and end at b , or cross e_2 and end at a . Similar conditions hold for edges that cross e_1 or e_2 by symmetry.

4-connectivity. In this case, the combinatorial characterization in Proposition 7.1 is more involved, and we need to maintain information for bundles of new edges that cross an old edge. For an edge $e \in E_0$, integer $q \in \{0, \dots, \ell\}$, vector $\vec{c} \in \{1, \dots, \ell\}^q$, and a laminar set system P over $\{1, \dots, q\}$, let $\text{OPT}_4(e, q, \vec{c}, P)$ denote the minimum size of a set of new edges F such that $G' = (V, E \cup F)$ has the following properties:

- Edge e crosses exactly q edges in F : $f_1, \dots, f_p$;
- for $i = 1, \dots, p$, edge $f_i \in F$ crosses at most c_i edges $h \in E_0$, $h \preceq e$;
- each set $S \in P$ corresponds to a set $F_{e,S} = \{f_i : i \in S\}$ such that the edges in $F_{e,S}$ have a common endpoint below e or $F_{e,S}$ is the set of all edges in F that cross an edge $h \preceq e$;
- every edge $h \in E_0$, $h \preceq e$, satisfies the conditions in Proposition 7.1 for $k = 4$ assuming the edges in each set $F_{e,S}$, $S \in P$, do not have a common endpoint *above* e .

If no such set F exists, then we set $\text{OPT}_4(e, q, \vec{c}, P) = \infty$.

Our DP algorithm considers all edges $e \in E_0$ in increasing order in the poset $(E_0, \preceq)$. If e is a minimal element of the poset $(E_0, \preceq)$, then e is an ear. In this case, all edges that cross e must have a common endpoint below e : We set $\text{OPT}_3(e, q, \vec{c}, P) = q$ for all $\vec{c}$ if $P = \{\{1, \dots, q\}\}$, and $\text{OPT}_3(e, q, \vec{c}, P) = \infty$ for any other partition P ; see Figure 12a. If e is not minimal, then $e = ab$ in a triangle $\Delta(a, b, c)$, where $e_1 = ac$ or $e_2 = bc$ is also a diagonal in E_0 . We compute $\text{OPT}_4(e, q, \vec{c}, P)$ by a brute force comparison of all possible subproblems for edges e_1 and e_2 . By maintaining the laminar system P , we can determine whether every non-ear edge is crossed by at least two disjoint edges. ◀

8 Augmentation for Planar Straight-Line Graphs

As noted in Section 1, the connectivity augmentation problem has been thoroughly studied over PSLGs, where both the input and output graphs are restricted to PSLGs (hence $k \leq 5$). For example, every PSLG G can be augmented to a 2-connected PSLG, and this is the best possible if the vertices are in convex position or an edge of G is a chord of the convex hull. In this section, we relax the planarity constraint for the output graph G' . In general, we would like to determine all pairs of parameters (k, ℓ) such that every PSLG can be augmented to a k -connected ℓ -planar geometric graph.

If we wish to augment a PSLG to a 3-connected geometric graph, we need to go beyond planarity. Our first result addresses the case $k = 3$ for points in convex position.

► **Theorem 8.1.** *Every PSLG on n points in convex position can be augmented to a 3-connected 5-planar geometric graph. This bound is the best possible: There exists a triangulation on n points in convex position for which any augmentation to a 3-connected geometric graph yields an edge with at least 5 crossings.*

We can generalize Theorem 8.1 and obtain an asymptotically tight trade-off between connectivity and local crossing number (which matches the bound in Proposition 2.2).

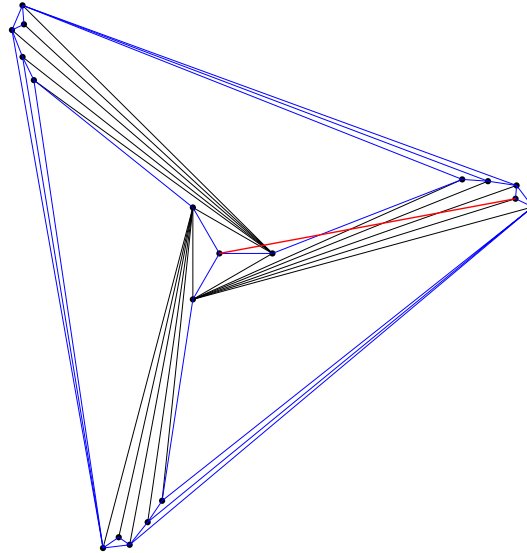
► **Theorem 8.2.** *For every $k \in \mathbb{N}$, every PSLG on n points in convex position can be augmented to a k -connected $O(k^2)$ -planar geometric graph; and this bound is the best possible.*

Proof. We may assume w.l.o.g. that the input is a triangulation G on n points in convex position. The dual graph is a binary tree with n nodes. We can partition the dual tree into subtrees (*clusters*) of size in the range $[k, 3k]$. Each cluster is a union of triangles, i.e., a convex polygon with $\Theta(k)$ vertices. The adjacency graph of the clusters (*cluster graph*) is a tree of maximum degree $O(k)$.

We can augment G as follows. (1) Augment each cluster to a complete graph; (2) for each pair of adjacent clusters V_1 and V_2 (where $|V_1 \cap V_2| = 2$) add a matching of size $k - 2$ between $V_1 \setminus V_2$ and $V_2 \setminus V_1$. Let G' be the augmented graph. Similarly to the proof of Theorem 3.2, the geometric graph G' is k -connected (by induction on the clusters), and $O(k^2)$ -planar (arguing about edges induced by a cluster, and a pair of adjacent clusters). ◀

For n points in general position, we cannot always augment a PSLG to 4-connectivity with bounded local crossing number.

► **Theorem 8.3.** *For every $n \in \mathbb{N}$, there is a straight-line triangulation G on n points such that any augmentation to a 4-connected geometric graph has an edge with $\Omega(n)$ crossings.*



■ **Figure 13** Construction in the proof of Theorem 8.3.

Proof. We construct a straight-line triangulation as follows; see Figure 13. Start with a PSLG K_4 , with three points at the vertices of a regular triangle, and one at the center. Attach $(n - 4)/3$ long edges to each outer vertex to form a “fan” almost parallel to the three sides of the regular triangle. Let G be an arbitrary triangulation of this graph. In any 4-connected augmentation G' of G , the central vertex of the initial K_4 must be connected to a new vertex. This edge will cross $\Omega(n)$ edges of one of the fans, hence $\Omega(n)$ edges of G . ◀

References

- 1 Manuel Abellanas, Alfredo García, Ferran Hurtado, Javier Tejel, and Jorge Urrutia. Augmenting the connectivity of geometric graphs. *Comput. Geom.*, 40(3):220–230, 2008. doi:10.1016/J.COMGEO.2007.09.001.

- 2 Anna Accornero, Massimo Ancona, and Sonia Varini. All separating triangles in a plane graph can be optimally "broken" in polynomial time. *Int. J. Found. Comput. Sci.*, 11(3):405–421, 2000. doi:10.1142/S012905410000020X.
- 3 Miklós Ajtai, Vašek Chvátal, Monroe M. Newborn, and Endre Szemerédi. Crossing-free subgraphs. In *Theory and Practice of Combinatorics*, volume 60 of *North-Holland Math. Stud.*, pages 9–12. North-Holland, 1982.
- 4 Hugo A. Akitaya, Rajasekhar Inkulu, Torrie L. Nichols, Diane L. Souvaine, Csaba D. Tóth, and Charles R. Winston. Minimum weight connectivity augmentation for planar straight-line graphs. *Theor. Comput. Sci.*, 789:50–63, 2019. doi:10.1016/J.TCS.2018.05.031.
- 5 Marwan Al-Jubeh, Mashhood Ishaque, Kristóf Rédei, Diane L. Souvaine, Csaba D. Tóth, and Pavel Valtr. Augmenting the edge connectivity of planar straight line graphs to three. *Algorithmica*, 61(4):971–999, 2011. doi:10.1007/S00453-011-9551-0.
- 6 Greg Aloupis, Luis Barba, Paz Carmi, Vida Dujmović, Fabrizio Frati, and Pat Morin. Compatible connectivity augmentation of planar disconnected graphs. *Discret. Comput. Geom.*, 54(2):459–480, 2015. doi:10.1007/S00454-015-9716-8.
- 7 Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994. doi:10.1145/174644.174650.
- 8 David W. Barnette and Brank Grünbaum. On Steinitz's theorem concerning convex 3-polytopes and on some properties of planar graphs. In G. Chartrand and S. F. Kapoor, editors, *The Many Facets of Graph Theory*, pages 27–40. Springer, 1969. doi:10.1007/BFb0060102.
- 9 Therese Biedl. A note on 1-planar graphs with minimum degree 7. *Discret. Appl. Math.*, 289:230–232, 2021. doi:10.1016/J.DAM.2020.10.004.
- 10 Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.
- 11 Hans L. Bodlaender and Ton Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21(2):358–402, 1996. doi:10.1006/JAGM.1996.0049.
- 12 Prosenjit Bose, Jurek Czyzowicz, Zhicheng Gao, Pat Morin, and David R. Wood. Simultaneous diagonal flips in plane triangulations. *J. Graph Theory*, 54(4):307–330, 2007. doi:10.1002/JGT.20214.
- 13 Prosenjit Bose, Dana Jansens, André Van Renssen, Maria Saumell, and Sander Verdonschot. Making triangulations 4-connected using flips. *Comput. Geom.*, 47(2):187–197, 2014. doi:10.1016/J.COMGEO.2012.10.012.
- 14 Prosenjit Bose and Sander Verdonschot. A history of flips in combinatorial triangulations. In *Abstracts of the XIV Spanish Meeting on Computational Geometry (EGC)*, volume 7579 of *LNCS*, pages 29–44. Springer, 2011. doi:10.1007/978-3-642-34191-5_3.
- 15 Aaron Büngener and Michael Kaufmann. Improving the crossing lemma by characterizing dense 2-planar and 3-planar graphs. In *Proc. 32nd Symposium on Graph Drawing and Network Visualization (GD)*, volume 320 of *LIPICs*, pages 29:1–29:22. Schloss Dagstuhl, 2024. doi:10.4230/LIPICs.GD.2024.29.
- 16 Jean Cardinal, Michael Hoffmann, Vincent Kusters, Csaba D. Tóth, and Manuel Wettstein. Arc diagrams, flip distances, and Hamiltonian triangulations. *Comput. Geom.*, 68:206–225, 2018. doi:10.1016/J.COMGEO.2017.06.001.
- 17 Ruoxu Cen, Jason Li, and Debmalya Panigrahi. Edge connectivity augmentation in near-linear time. In *Proc. 54th ACM Symposium on Theory of Computing (STOC)*, pages 137–150, 2022. doi:10.1145/3519935.3520038.
- 18 S. Dhanalakshmi, N. Sadagopan, and V. Manogna. Tri-connectivity augmentation in trees. *Electron. Notes Discret. Math.*, 53:57–72, 2016. doi:10.1016/J.ENDM.2016.05.006.
- 19 Kapali P. Eswaran and Robert Endre Tarjan. Augmentation problems. *SIAM J. Comput.*, 5(4):653–665, 1976. doi:10.1137/0205044.

- 20 Sergej Fialko and Petra Mutzel. A new approximation algorithm for the planar augmentation problem. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 260–269, 1998. URL: <http://dl.acm.org/citation.cfm?id=314613.314714>.
- 21 András Frank and Tibor Jordán. Graph connectivity augmentation. In *Handbook of Graph Theory, Combinatorial Optimization, and Algorithms*, chapter 14, pages 313–346. CRC Press, 2015.
- 22 Greg N. Frederickson and Joseph F. JáJá. Approximation algorithms for several graph augmentation problems. *SIAM J. Comput.*, 10(2):270–283, 1981. doi:10.1137/0210019.
- 23 Jun Fujisawa, Keita Segawa, and Yusuke Suzuki. The matching extendability of optimal 1-planar graphs. *Graphs Comb.*, 34(5):1089–1099, 2018. doi:10.1007/S00373-018-1932-6.
- 24 Jérôme Galtier, Ferran Hurtado, Marc Noy, Stephane Perennes, and Jorge Urrutia. Simultaneous edge flipping in triangulations. *Int. J. Comput. Geom. Appl.*, 13(2):113–133, 2003. doi:10.1142/S0218195903001098.
- 25 Alfredo García, Ferran Hurtado, Clemens Huemer, Javier Tejel, and Pavel Valtr. On triconnected and cubic plane graphs on given point sets. *Comput. Geom.*, 42(9):913–922, 2009. doi:10.1016/J.COMGEO.2009.03.005.
- 26 Alfredo García, Ferran Hurtado, Marc Noy, and Javier Tejel. Augmenting the connectivity of outerplanar graphs. *Algorithmica*, 56(2):160–179, 2010. doi:10.1007/S00453-008-9167-1.
- 27 Carsten Gutwenger, Petra Mutzel, and Bernd Zey. On the hardness and approximability of planar biconnectivity augmentation. In *Proc. 15th Conference on Computing and Combinatorics (COCOON)*, volume 5609 of *LNCS*, pages 249–257. Springer, 2009. doi:10.1007/978-3-642-02882-3_25.
- 28 Carsten Gutwenger, Petra Mutzel, and Bernd Zey. Planar biconnectivity augmentation with fixed embedding. In *Proc. 20th International Workshop on Combinatorial Algorithms, (IWOCA)*, volume 5874 of *LNCS*, pages 289–300. Springer, 2009. doi:10.1007/978-3-642-10217-2_29.
- 29 Tsan-sheng Hsu. On four-connecting a triconnected graph. *J. Algorithms*, 35(2):202–234, 2000. doi:10.1006/JAGM.2000.1077.
- 30 Dávid Hudák, Tomáš Madaras, and Yusuke Suzuki. On properties of maximal 1-planar graphs. *Discussiones Mathematicae Graph Theory*, 32:737–747, 2012. doi:10.7151/dmgt.1639.
- 31 Ferran Hurtado and Csaba D. Tóth. Plane geometric graph augmentation: a generic perspective. In János Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 327–354. Springer, 2013. doi:10.1007/978-1-4614-0110-0_17.
- 32 Bill Jackson and Tibor Jordán. Independence free graphs and vertex connectivity augmentation. *J. Comb. Theory B*, 94(1):31–77, 2005. doi:10.1016/J.JCTB.2004.01.004.
- 33 Kasper Skov Johansen, Eva Rotenberg, and Carsten Thomassen. Edge-connectivity augmentation of simple graphs. *SIAM J. Discret. Math.*, 39(1):163–169, 2025. doi:10.1137/23M1574245.
- 34 Tibor Jordán. On the optimal vertex-connectivity augmentation. *J. Comb. Theory, Ser. B*, 63(1):8–20, 1995. doi:10.1006/JCTB.1995.1002.
- 35 Goos Kant. Augmenting outerplanar graphs. *J. Algorithms*, 21(1):1–25, 1996. doi:10.1006/JAGM.1996.0034.
- 36 Goos Kant. A more compact visibility representation. *International Journal of Computational Geometry & Applications*, 07(03):197–210, 1997. doi:10.1142/S0218195997000132.
- 37 Goos Kant and Hans L. Bodlaender. Planar graph augmentation problems. In *Proc. 2nd Workshop on Algorithms and Data Structures (WADS)*, volume 519 of *LNCS*, pages 286–298. Springer, 1991. doi:10.1007/BFB0028270.
- 38 Ken-ichi Kawarabayashi and Kenta Ozeki. 4-connected projective-planar graphs are Hamiltonian-connected. *J. Comb. Theory B*, 112:36–69, 2015. doi:10.1016/J.JCTB.2014.11.006.
- 39 Ton Kloks. *Treewidth: Computations and Approximations*, volume 842 of *LNCS*. Springer, 1994. doi:10.1007/BFB0045375.

- 40 Frank Thomson Leighton. *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-Exchange Graph and Other Networks*. MIT Press, 1983. URL: <https://mitpress.mit.edu/9780262621786/complexity-issues-in-vlsi/>.
- 41 David Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982. doi:10.1137/0211025.
- 42 Ryuichi Mori, Atsuhiko Nakamoto, and Katsuhiko Ota. Diagonal flips in hamiltonian triangulations on the sphere. *Graphs Comb.*, 19(3):413–418, 2003. doi:10.1007/S00373-002-0508-6.
- 43 Hiroshi Nagamochi and Peter Eades. Edge-splitting and edge-connectivity augmentation in planar graphs. In *Proc. 6th Integer Programming and Combinatorial Optimization (IPCO)*, volume 1412 of *LNCS*, pages 96–111. Springer, 1998. doi:10.1007/3-540-69346-7_8.
- 44 Kenta Ozeki and Carol T. Zamfirescu. Every 4-connected graph with crossing number 2 is Hamiltonian. *SIAM J. Discret. Math.*, 32(4):2783–2794, 2018. doi:10.1137/17M1138443.
- 45 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17:427–439, 1997. doi:10.1007/BF01215922.
- 46 David Rappaport. Computing simple circuits from a set of line segments is NP-complete. *SIAM J. Comput.*, 18(6):1128–1139, 1989. doi:10.1137/0218075.
- 47 Ignaz Rutter and Alexander Wolff. Augmenting the connectivity of planar and geometric graphs. *J. Graph Algorithms Appl.*, 16(2):599–628, 2012. doi:10.7155/JGAA.00275.
- 48 Yusuke Suzuki. Re-embeddings of maximum 1-planar graphs. *SIAM J. Discret. Math.*, 24(4):1527–1540, 2010. doi:10.1137/090746835.
- 49 Robin Thomas and Xingxing Yu. 4-connected projective-planar graphs are Hamiltonian. *J. Comb. Theory B*, 62(1):114–132, 1994. doi:10.1006/JCTB.1994.1058.
- 50 Csaba D. Tóth. Connectivity augmentation in planar straight line graphs. *European J. Combinatorics*, 33:408–425, 2012. doi:10.1016/j.ejc.2011.09.002.
- 51 Vera Traub and Rico Zenklusen. Local search for weighted tree augmentation and Steiner tree. In *Proc. 33rd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3253–3272, 2022. doi:10.1137/1.9781611977073.128.
- 52 William T. Tutte. A theorem on planar graphs. *Trans. Amer. Math. Soc.*, 82:99–116, 1956. doi:10.1090/s0002-9947-1956-0081471-8.
- 53 László A. Végh. Augmenting undirected node-connectivity by one. *SIAM J. Discret. Math.*, 25(2):695–718, 2011. doi:10.1137/100787507.
- 54 Toshimasa Watanabe and Akira Nakamura. Edge-connectivity augmentation problems. *J. Comput. Syst. Sci.*, 35(1):96–144, 1987. doi:10.1016/0022-0000(87)90038-9.
- 55 Toshimasa Watanabe and Akira Nakamura. A minimum 3-connectivity augmentation of a graph. *J. Comput. Syst. Sci.*, 46(1):91–127, 1993. doi:10.1016/0022-0000(93)90050-7.