Augmenting the Connectivity of Planar and Geometric Graphs

Ignaz Rutter* Alexander Wolff†

Abstract

In this paper we study some connectivity augmentation problems. Given a connected graph G with some property $\mathcal{P}$, we want to make G 2-vertex connected (or 2-edge connected) by adding edges such that the resulting graph keeps property $\mathcal{P}$. The aim is to add as few edges as possible. The property that we consider is planarity, both in an abstract graph-theoretic and in a geometric setting.

We show that it is NP-hard to find a minimum-cardinality augmentation that makes a planar graph 2-edge connected. For making a planar graph 2-vertex connected this was known. We further show that both problems are hard in the geometric setting, even when restricted to trees. On the other hand we give polynomial-time algorithms for the special case of convex geometric graphs.

We also study the following related problem. Given a plane geometric graph G, two vertices s and t of G, and an integer k, how many edges have to be added to G such that G contains k edge- (or vertex-) disjoint s–t paths? For $k = 2$ we give optimal worst-case bounds; for $k = 3$ we characterize all cases that have a solution.

1 Introduction

Augmenting a given graph to increase its connectivity is important, e.g., for securing communication networks against node and link failures. The planar version of the problem, where the augmentation has to preserve planarity, also has applications in graph drawing [8]. Many graph-drawing algorithms guarantee nice properties (such as convex faces) for graphs with high connectivity. To apply such an algorithm to a less highly connected graph, one adds edges until one reaches the required level of connectivity, uses the algorithm to produce the drawing, and finally removes the added edges again. However, with each removal of an edge one might lose some of the nice properties (such as the convexity of a face). Hence it is natural to look for an augmentation that uses as few edges as possible. Recall that a graph is *k-vertex connected* (*k-edge connected*) if the removal of any subset of $k-1$ vertices (edges) does not make the graph disconnected.

We consider the following two problems.

Planar 2-Vertex Connectivity Augmentation (PVCA):

> Given a connected planar graph $G = (V, E)$ with $n := |V|$ and $m := |E|$, find a smallest set E' of vertex pairs such that the graph $G' = (V, E \cup E')$ is planar and 2-vertex connected (biconnected).

Planar 2-Edge Connectivity Augmentation (PECA) is defined as PVCA, but with *2-vertex connected* replaced by *2-edge connected* (bridge-connected).

The corresponding problems without the planarity constraints have a long history, both for directed and undirected graphs. The unweighted cases can be solved in polynomial time, while the weighted versions are hard [2]. Frederickson and Ja'Ja' [4] gave $O(n^2)$-time factor-2 approximations and showed that augmenting a directed acyclic graph to be strongly connected, and augmenting a tree to be bridge- or biconnected, is NP-complete—even if weights are restricted to the set $\{1, 2\}$. Hsu [5] gave an $O(m + n)$-time algorithm for (unit-weight) 2-vertex connectivity augmentation.

Kant and Bodlaender [8] showed that PVCA is NP-complete and gave 2-approximations for both PVCA and PECA that run in $O(n \log n)$ time. Their 1.5-approximation turned out to be wrong [3]. Fialko and Mutzel gave a 5/3-approximation [3]. Kant showed that PVCA and PECA can be solved in linear time for outerplanar graphs [7].

Provan and Burk [10] considered related problems. Given a planar graph $G = (V, E_G)$ and a planar biconnected (bridge-connected) graph $H = (V, E_H)$ with $E_G \subseteq E_H$, find a smallest set $E' \subseteq E_H$ such that $G' = (V, E_G \cup E')$ is planar and biconnected (bridge-connected). They show that both problems are NP-hard if G is not necessarily connected and give $O(n^4)$-time algorithms for the connected cases.

We also consider a geometric version of the above problems. Recall that a *geometric* graph is a graph where each vertex v corresponds to a point $\mu(v)$ in the plane and where each edge uv corresponds to the straight-line segment $\overline{\mu(u)\mu(v)}$. We are exclusively

*Fakultät für Informatik, Universität Karlsruhe, P.O. Box 6980, D-76128 Karlsruhe, Germany. Supported by grant WO 758/4-3 of the German Science Foundation (DFG). WWW: `i11www.ira.uka.de/people/rutter`

†Faculteit Wiskunde en Informatica, Technische Universiteit Eindhoven, WWW: `www.win.tue.nl/~awolff`

problem	planar	outerplanar	geometric	convex
PVCA	NPC [8]	$O(n)$ [7]	NPC	$O(n)$
PECA	NPC	$O(n)$ [7]	NPC	$O(n)$
weighted PVCA	NPC	open	NPC	$O(n^2)$
weighted PECA	NPC	open	NPC	$O(n)$

Table 1: Complexity of PVCA and PECA.

interested in geometric graphs that are *plane*, that is, whose edges intersect at most in their endpoints. Therefore, in this paper by geometric graph we always mean a plane geometric graph. Given a geometric graph G we again want to find a (small) set of vertex pairs such that adding the corresponding edges to G leaves G plane and augments its connectivity.

Rappaport [11] has shown that it is NP-complete to decide whether a set of line segments can be connected to a simple polygon, i.e., geometric PVCA and PECA are NP-complete. Abellanas et al. [1] have shown worst-case bounds for geometric PVCA and PECA. For geometric PVCA they show that $n-2$ edges are sometimes needed and are always sufficient. For geometric PECA they prove that $2n/3$ edges are sometimes needed and $6n/7$ edges are always sufficient. In the special case of plane geometric trees they show that $n/2$ edges are sometimes needed and that $2n/3$ edges are always sufficient for PECA.

Our results. First we show that PECA is NP-complete, too. This answers an open question posed by Kant [6].

Second, we sharpen the result of Rappaport [11] by showing that geometric PVCA and PECA are NP-complete even if restricted to trees.

Third, we give algorithms that solve geometric PVCA and PECA in polynomial time for *convex* geometric graphs, that is, graphs whose vertex sets correspond to point sets in convex position.

Table 1 gives an overview about our results and what has been known previously about the complexity of PVCA and PECA.

Fourth, we consider a related problem, the geometric s–t path augmentation problem. Given a plane geometric graph G, two vertices s and t of G, and an integer $k > 0$, is it possible to augment G such that it contains k edge-disjoint (k vertex-disjoint) s–t paths? We restrict ourselves to $k \in \{2, 3\}$. For $k = 2$ we show that edge-disjoint s–t path augmentation can always be done and needs at most $n/2$ edges. We give an algorithm that computes such an augmentation in linear time. The tree that yields the above-mentioned lower-bound of Abellanas et al. [1] also shows that our bound is tight. For $k = 3$ we show that edge-disjoint s–t path augmentation is always possible, and we give an $O(n^2)$-time algorithm that decides whether a given graph has a vertex-disjoint s–t path augmentation.

2 Complexity results

In this section we show that PECA is NP-complete. This settles an open problem posed by Kant and Bodlaender [8]. Our proof also implies that PVCA is NP-complete, which was already shown by Kant and Bodlaender [8].

Theorem 1 PECA *is NP-complete.*

The hardness proof is by reduction from PLANAR3SAT, which is known to be NP-hard [9]. The main idea is to use a base graph that is 3-connected (and hence has a unique embedding) and to add some leaves (i.e., degree-1 vertices) to this graph. These leaves can then be embedded in different faces of the graph. It is clear that in order to increase the connectivity the degree of each of the leaves must be enlarged. Ideally (i.e., if the given planar 3SAT formula is satisfiable) the embedding is chosen in such a way that the number of leaves in each face is even, because in this case we need only one edge for every two leaves, which is optimal.

Now we consider geometric PVCA and geometric PECA for connected graphs. These problems are NP-complete as well, however for reasons very different from the planar case. In the geometric setting the embedding is fixed, but two leaves lying in the same face cannot necessarily be connected by a straight-line segment without violating planarity. Especially adding one edge can rule out several others. For example in a square one could add one of the diagonals, but not both. This can again be used to construct a reduction from PLANAR3SAT.

Theorem 2 *Let G be a plane geometric graph and $k > 0$ be an integer. It is NP-complete to decide whether adding k edges suffices to make G bridge- or biconnected. This is true even if G has exactly $2k$ leaves and G is a tree.*

Once we have shown the result for connected graphs it is easy to extend this to trees. We reduce from the previous case. Let G be a connected plane geometric graph. As long as G contains a cycle, replace an arbitrary edge of a cycle by the construction shown in Figure 1. Call the resulting tree T. Clearly an optimal augmentation connects the two leaves of the construction. Hence an optimal augmentation of T induces an optimal augmentation of G.

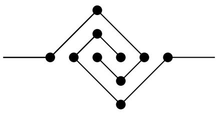

Figure 1: Construction for removing cycles in G.

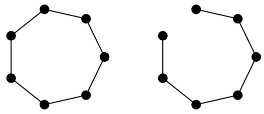

Figure 2: A cycle (left) and a near-cycle (right).

3 Convex geometric graphs

In this section we consider the geometric version of PVCA and PECA in the special case that the input graph is a convex geometric graph. We call an edge *outer edge* if it belongs to the convex hull and *inner edge* otherwise.

Note that PVCA for a convex geometric graph G is trivial: G is biconnected if and only if it contains all edges of the convex hull. Thus we focus on PECA.

If a connected convex geometric graph does not contain an inner edge then it is either a *cycle* or a *near-cycle*, see Figure 2. While the cycle is already bridge-connected, we need a single edge to make the near-cycle bridge-connected.

The basic idea is to decompose an arbitrary convex geometric graph into cycles and near-cycles and use this decomposition to compute an edge set of minimum cardinality that bridge-connects the graph.

Given a convex geometric graph $G = (S, E)$ and an inner edge e of G, we define an operation that we call *splitting* G *at* e. Splitting G at e yields two subgraphs G_e^+ and G_e^- of G induced by the vertices to the left of or on e and to the right of or on e, respectively.

Lemma 3 *Let G be a connected convex geometric graph and let f be an outer edge of G. If G is not a cycle or a near-cycle, then there exists an inner edge e such that G_e^+ or G_e^- is a cycle or a near-cycle that does not contain f.*

We use Lemma 3 to repeatedly cut off (near-) cycles (i.e., remove all its edges except the split-edge) from G. Each time we cut off a near-cycle, we add to our augmentation the edge that completes the cycle. This is optimal since no edge can cross the split-edge and hence both sides can be processed independently.

If an endpoint of the split-edge is a leaf, we can also remove the split-edge. Otherwise we mark the split-edge and remove it as soon as one of its endpoints becomes a leaf. This ensures that every near-cycle that is cut off actually requires an additional edge in the augmentation. Let's summarize the above.

Theorem 4 *Let $S \subset \mathbb{R}^2$ be a set of n points and let $G = (S, E)$ be a connected convex geometric graph. If the convex hull of S and the corresponding embedding of G are given, we can compute in $O(n)$ time and space a set E' of vertex pairs of minimum cardinality such that $G' = (S, E \cup E')$ is bridge-connected.*

By combining the previous approach with dynamic programming we can also solve the case where each pair of vertices has a positive weight and the aim is to minimize the total weight of the augmentation. Time and space consumption become quadratic.

4 s–t path augmentation

In this section we consider the following problems: Given a plane geometric graph $G = (S, E)$, two vertices $s \neq t$ of G, and an integer $k > 0$, find a smallest set E' of vertex pairs such that $G' = (S, E \cup E')$ is plane and contains k edge-disjoint s–t paths. We also consider the corresponding problem for vertex-disjoint s–t paths. We treat the cases $k = 2$ and $k = 3$.

4.1 Path augmentation for k = 2

The case $k = 2$ is a relaxed version of PECA and PVCA. Although the problem is very restricted in comparison to full 2-edge or 2-vertex connectivity augmentation, it does not seem to be much easier. Like Abellanas et al. [1] we consider the corresponding worst-case problem: how many edges are needed for an s–t path augmentation in the worst case?

Let's quickly discuss the vertex-disjoint case. For the lower bound we can re-use the example of Abellanas et al. [1]: a zig-zag path with end vertices s and t whose vertices are in convex position. There, $n - 2$ edges are needed to establish two vertex-disjoint s–t paths. On the other hand it is not hard to see that $n - 2$ edges always suffice.

Now let's turn to the more interesting edge-disjoint case. Here, the zig-zag path yields a lower bound of $n/2$. Note that the solution actually makes the graph bridge-connected. In fact, Abellanas et al. [1] conjecture that any geometric n-vertex tree can be made bridge-connected by adding at most $n/2$ edges. We show that there is always an s–t path augmentation with at most $n/2$ edges, which is tight by the zig-zag example. We also give a simple algorithm that finds such an augmentation in linear time.

Lemma 5 *Let $S \subseteq \mathbb{R}^2$, let $G = (S, E)$ be a connected plane geometric graph, and let $G' = (S, E')$ with $E \subseteq E'$ be any plane geometric graph that contains G. If s and t are two vertices of G, and G' contains a path of length ℓ between s and t, then there exists an s–t path augmentation of G with at most ℓ edges.*

The proof shows how the path in G' can be used to determine an augmentation for G. The most interesting case is that the path in G' uses an edge that actually is a bridge b in G. The crucial step is to show that we can add a suitable edge to G that induces a cycle with b on it, i.e., b is no longer a bridge. In all other cases we either do not need to add an edge, or we can use the edge of the path.

Let $G = (S, E)$ be a geometric graph. A *triangulation of* G is a triangulation $T = (S, E')$ of the convex hull of S with $E \subseteq E'$. It is well known that every geometric graph can be triangulated [1]. We show:

Lemma 6 *Let $S \subset \mathbb{R}^2$ be a set of n points and let $T = (S, E)$ be a triangulation of the convex hull of S. Then the diameter of T is at most $n/2$.*

The basic idea is to consider growing neighborhoods of the vertices s and t. The *i-th iterated neighborhood* $N_i(v)$ of a vertex v contains all vertices of G within (graph-theoretic) distance at most i from v. Let k be the smallest integer such that $N_k(s) \cap N_k(t) \neq \emptyset$. We use a lower bound on the size of i-th iterated neighborhoods and the fact that $N_{k-1}(s) \cap N_{k-1}(t) = \emptyset$ in order to obtain an upper bound on k. Since G contains an s–t path of length at most $2k$, this upper bound implies the claim.

Lemmas 5 and 6 immediately yield the following.

Theorem 7 *Let $S \subset \mathbb{R}^2$, let $G = (S, E)$ be any plane connected geometric graph with n vertices, and let s and t be any two vertices of G. Then there always exists a set E' of at most $n/2$ pairs of points in S such that $G' = (S, E \cap E')$ is again a plane geometric graph and contains two edge-disjoint s–t paths. Such a set can be computed in linear time.*

This bound can be improved if the convex hull of S does not contain too many points. Take any triangulation and consider the iterated neighborhoods of s and t. As long as a neighborhood has not reached the convex hull it grows by at least three vertices with every iteration. As soon as both neighborhoods have reached the convex hull, we can connect them by a path along the convex hull.

Lemma 8 *The diameter of a plane triangulation $T = (S, E)$ is at most $2(n + 3)/5 + h/2$, where $n = |S|$ and h is the number of vertices on the convex hull of S.*

This improves Lemma 6 for $h < (n - 12)/5$.

4.2 Path augmentation for $k = 3$

We first consider the problem of finding three *vertex*-disjoint s–t paths in a geometric graph. Let $T = (S, E)$ be any plane geometric triangulation and let s and t be any two vertices of T. An edge connecting two vertices of the convex hull that does not belong to the convex hull itself is called a *chord*. A chord uv is (s, t)*-separating* if s and t are in different connected components of $T \setminus uv$.

Obviously T has three vertex-disjoint s–t paths if and only if T does not contain an (s, t)-separating chord. Hence we can rephrase our original question as follows: does any plane geometric graph G without (s, t)-separating chords have a triangulation without (s, t)-separating chords? Such a triangulation would then contain the desired augmentation.

Theorem 9 *Let $S \subset \mathbb{R}^2$ be a set of n points, let $G = (S, E)$ be a connected plane geometric graph, and let $s \neq t$ be two vertices of G. If G contains no (s, t)-separating chord, then there is a triangulation T_G of G that contains three vertex-disjoint s–t paths. Such a triangulation can be computed in $O(n^2)$ time.*

Now we consider the problem of finding three *edge*-disjoint s–t paths. For triangulations we have the following characterization.

Theorem 10 *Let $T = (S, E)$ be a triangulation of the convex hull of S and let $s, t \in S$. Then T contains three edge-disjoint s–t paths if and only if s and t have degree at least 3.*

It is easy to check whether a given geometric graph can be triangulated in such a way.

References

[1] M. Abellanas, A. García, F. Hurtado, J. Tejel, and J. Urrutia. Augmenting the connectivity of geometric graphs. *Comput. Geom. Theory Appl.*, 2008. Appeared online at `http://dx.doi.org/10.1016/j.comgeo.2007.09.001`.

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

[3] S. Fialko and P. Mutzel. A new approximation algorithm for the planar augmentation problem. In *Proc. 9th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA'98)*, pages 260–269, 1998.

[4] G. N. Frederickson and J. Ja'Ja'. Approximation algorithms for several graph augmentation problems. *SIAM J. Comput.*, 10(2):270–283, 1981.

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

[6] G. Kant. *Algorithms for Drawing Planar Graphs.* PhD thesis, University of Utrecht, 1993.

[7] G. Kant. Augmenting outerplanar graphs. *J. Algorithms*, 21(1):1–25, 1996.

[8] G. Kant and H. L. Bodlaender. Planar graph augmentation problems. In F. Dehne, J.-R. Sack, and N. Santoro, editors, *Proc. 2nd Workshop Algorithms and Data Structures (WADS'91)*, volume 519 of *Lecture Notes Comput. Sci.*, pages 286–298. Springer-Verlag, 1991.

[9] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[10] J. S. Provan and R. C. Burk. Two-connected augmentation problems in planar graphs. *J. Algorithms*, 32:87–107, 1999.

[11] D. Rappaport. Computing simple circuits from a set of line segments is NP-complete. *SIAM J. Comput.*, 18(6):1128–1139, 1989.