Distance Measures for Embedded Graphs - Revisited *

Hugo A. Akitaya[†1], **Maike Buchin**[2], **and Bernhard Kilgus**[‡3]

1 **Department of Computer Science, Tufts University, Medford, MA, USA**
hugo.alves_akitaya@tufts.edu
2 **Department of Computer Science, Technical University Dortmund, Dortmund, Germany**
Maike.Buchin@tu-dortmund.de
3 **Department of Mathematics, Ruhr-University Bochum, Bochum, Germany**
Bernhard.Kilgus@rub.de

Abstract

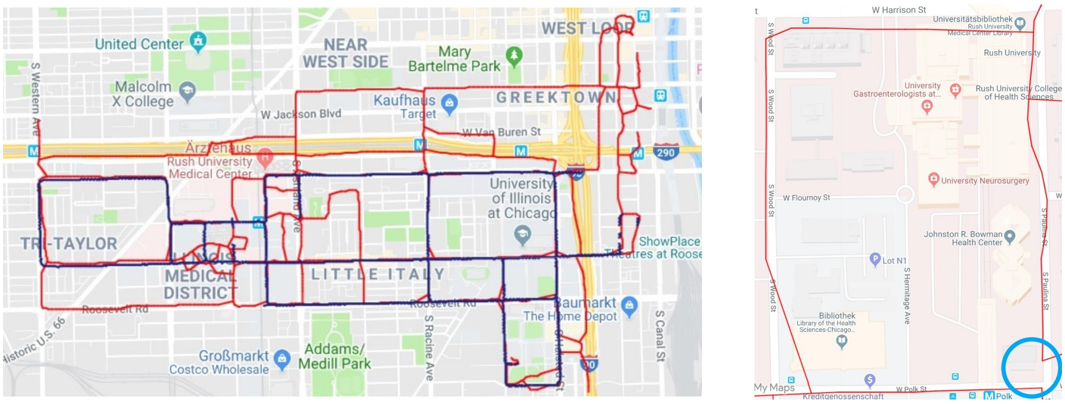
In this extended abstract, we present new hardness and algorithmic results for the graph distances presented at EuroCG 2017 [10]. We consider the case of the graph distance based on the Fréchet distance for plane graphs. We prove that deciding this distance is NP-hard and show how our general algorithmic approach yields an exact exponential time algorithm and a polynomial time approximation algorithm for this case.

* Full version available on arXiv [6].
† Supported by National Science Foundation grants CCF-1422311 and CCF-1423615, and the Science Without Borders scholarship program.
‡ Supported by the Deutsche Forschungsgemeinschaft (DFG), project BU 2419/3-1

35th European Workshop on Computational Geometry, Utrecht, The Netherlands, March 18–20, 2019.

1 Introduction

Motivation We study the task of comparing two embedded graphs. There are many applications that work with graphs embedded in an Euclidean space, such as road networks. For instance, by comparing two road networks one can assess the quality of map construction algorithms [3, 4], see Figure 1.



(a) Two partial map reconstructions of Chicago.

(b) Different topology.

Figure 1 Figure **(a)** shows the results of two map construction algorithms (blue: reconstuction by Davies et al. [12]; red: reconstruction by Ahmed et al. [5]). An appropriate measure for assessing the quality of the reconstruction should compare both the geometry and the topology of the reconstructions and the ground truth.

Related Work A few different approaches have been proposed for comparing such graphs. These are subgraph-isomorphism, edit distance [11], algorithms that compare all paths [1] or random samples of shortest paths [13], and the local persistent homology distance [2]. However, most of these capture only the geometry or only the topology of the embedded graphs. The sampling-based distance presented in [9] captures both, but it is not a formally defined distance. The traversal distance [7] is similar to the measures proposed here but captures the combinatorial structure of the graphs to a lesser extent.

Definitions and Previous Results Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two undirected graphs with vertices embedded as points in $\mathbb{R}^d$ (typically in the plane) that are connected by straight-line edges. We consider a mapping $s : G_1 \to G_2$ that maps each vertex $v \in V_1$ to a point $s(v)$ on G_2 (not necessarily a vertex) and that maps each edge $\{u, v\} \in E_1$ to a simple path in G_2 with endpoints $s(u)$ and $s(v)$. Our graph distances are generalizations of the (weak) Fréchet distance, popular distance measures for curves [8], to graphs: We define the *directed (weak) graph distance* $\vec{\delta}_{(w)G}$ as

$$\vec{\delta}_{(w)G}(G_1, G_2) = \inf_{s:G_1\to G_2} \max_{e\in E_1} \delta_{(w)F}(e, s(e)),$$

where $\delta_{(w)F}$ denotes the (weak) Fréchet distance, s ranges over all graph mappings from G_1 to G_2, and e and its image $s(e)$ are interpreted as curves in the plane.

The general algorithm to compute the directed (weak) graph distances is based on the definition of valid ε-placements of the vertices and edges. An *ε-placement* of a vertex v is a maximally connected component of G_2 restricted to the ε-ball $B_\varepsilon(v)$ around v. A (weak)

ε-placement of an edge $e = \{u, v\} \in E_1$ is a path P in G_2 with endpoints on ε-placements C_u of u and C_v of v such that $\delta_{(w)F}(e, P) \le \varepsilon$. In that case, we say that C_u and C_v are *reachable* from each other. An ε-placement C_v of v is *(weakly) valid* if for every neighbor u of v there exists an ε-placement C_u of u such that C_v and C_u are *reachable* from each other.

Deciding the directed (weak) graph distance is NP-hard for general graphs, but we can compute the (weakly) valid ε-placements in polynomial time [6, 10]. If there is a vertex with no (weakly) valid ε-placement it follows that $\vec{\delta}_{(w)G}(G_1, G_2) > \varepsilon$. Conversely, the existence of a (weakly) valid ε-placement for each vertex ensures $\vec{\delta}_{(w)G}(G_1, G_2) \le \varepsilon$ for several cases, namely if G_1 is a tree (both graph distances) and if G_1 and G_2 are plane graphs (weak graph distances). Therefore, the distances are decidable in polynomial time in these cases.

New Results In this paper, we show that deciding whether $\vec{\delta}_G(G_1, G_2) \le \varepsilon$ remains NP-hard if G_1 and G_2 are plane graphs, that is the existence of a valid ε-placement for each vertex is not a sufficient criterion for $\vec{\delta}_G(G_1, G_2) \le \varepsilon$ here. Furthermore, we prove an inapproximability result for this case. Subsequently, we present an exact exponential time algorithm and a polynomial time approximation algorithm based on the general algorithmic approach.

2 Hardness Results

▶ **Theorem 1.** *For plane graphs G_1, G_2, deciding whether $\vec{\delta}_G(G_1, G_2) \le \varepsilon$ is NP-hard.*

Proof. Here, we give a concise version of the proof. For a more elaborated version, see [6]. We prove the NP-hardness by a reduction from Monotone-Planar-3-Sat (MP3S). That is, we construct straight-line embedded graphs G_1, G_2 based on a MP3S instance A, with edges of G_2 labeled True or False. We describe the construction of the subgraphs (gadgets) for the Variables and Clauses of A and prove which binary combinations can be realized such that all edges and their images are within Fréchet distance at most ε. Figure 2 and 3 illustrate the gadgets and a partial graph construction. We denote the ε-tube around the edge e by $T_\varepsilon(e) = e \bigoplus B_\varepsilon$. A path labeled True (False) is shortly denoted as True (False) signal. All vertices of the graph can be either placed arbitrarily within a given ε-surrounding or must lie at the intersection of two lines. This ensures that the construction uses rational coordinates only and can be computed in polynomial time.

For the Variable gadget, we draw two edges, e_1, e_2, of G_1 in a $90° - 120°$ angle incident to a vertex v and add vertices w_1 (w_2) of G_2 at the intersection of the outer boundary of $T_\varepsilon(e_2)$ ($T_\varepsilon(e_1)$) and a line through e_1 (e_2). Furthermore, we add a vertex w_3 of G_2 at the intersection of the boundaries of $T_\varepsilon(e_1)$ and $T_\varepsilon(e_2)$. We connect w_1 and w_2 with w_3 and draw an edge from w_1 and w_2 inside the ε-tubes around e_1 and e_2, labeled True. Analogously, we place two edges from w_3 labeled False. For the Variable gadget a True-True combination is not possible: The vertex v has two placements p_1 and p_2. Assume we choose p_1. Then, one can map e_1 to a path containing the edge of G_2 with the True labeling inside $T_\varepsilon(e_1)$. Now, we want to map e_2 to a path P starting at some point of p_1, where P contains the edge of G_2 with the True labeling inside $T_\varepsilon(e_2)$. Thus P must contain w_3 and w_1. As $\delta_F(e_2, P) > \varepsilon$ (here, we only have $\delta_{wF}(e_2, P) \le \varepsilon$) for any such path P, this labeling is not realizable. It is easy to see that any other labeling of paths e_1 and e_2 are mapped to is realizable.

A Permute gadget is a differently labeled Variable gadget. For the Split gadget, we add a third edge e_3 of G_1 to the Variable gadget and add edges of G_2 from w_2 and w_3 inside the ε-tube around e_3. For the labeling, see Figure 2. A False signal can not be converted to a True signal in Split gadget or in the Permute gadget. However, a True signal can but does not need to be converted to a False signal in the Permute gadget.

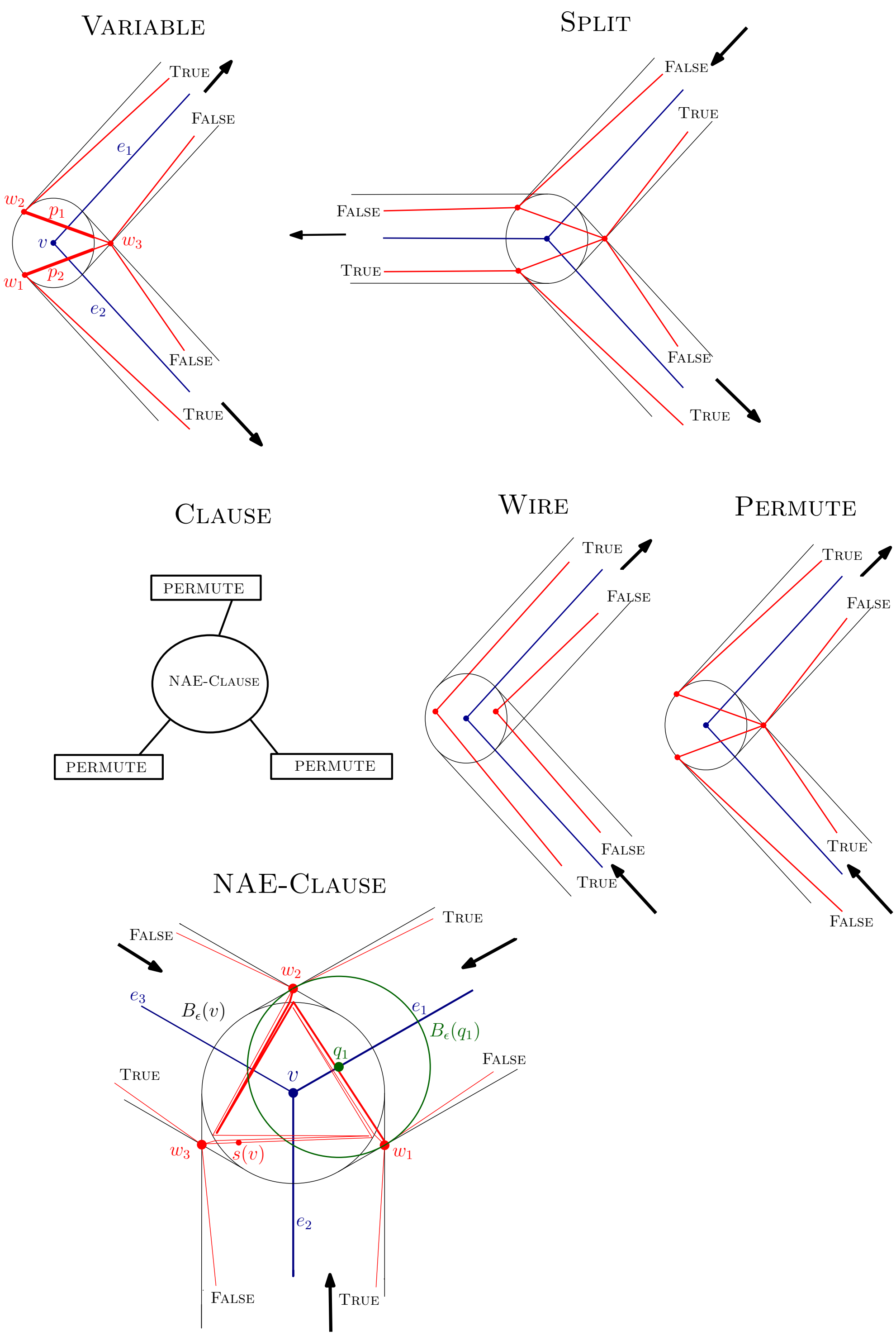


Figure 2 Gadgets to build a graph-similarity instance given a MONOTONE-PLANAR-3-SAT instance.

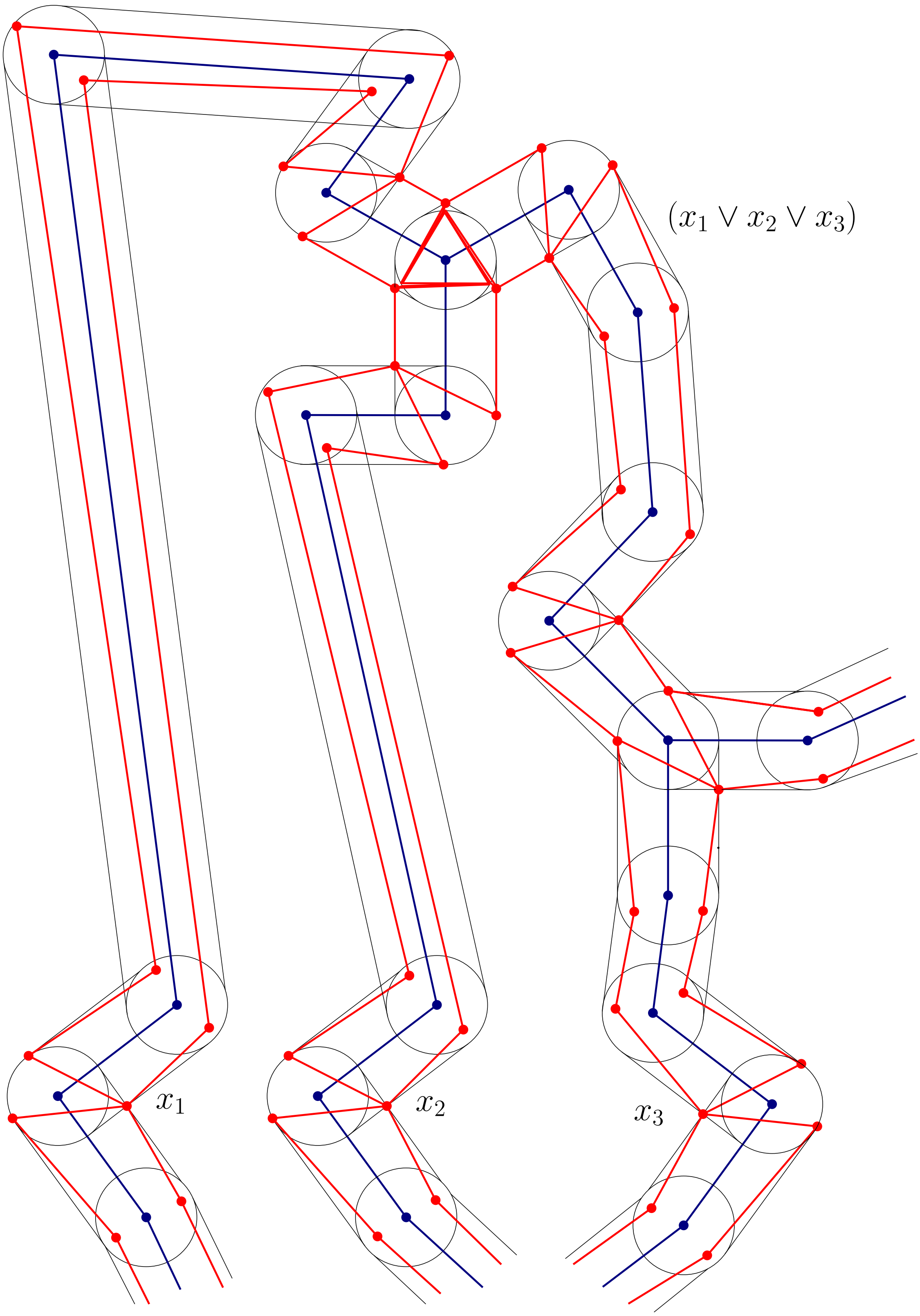


Figure 3 Construction of one CLAUSE gadget given the MP3S instance A with variables $V = \{x_1, x_2, \ldots, x_5\}$ and clauses $C = \{(x_1 \vee x_2 \vee x_3), (x_3 \vee x_4 \vee x_5), (\bar{x_1} \vee \bar{x_3} \vee \bar{x_5})\}$

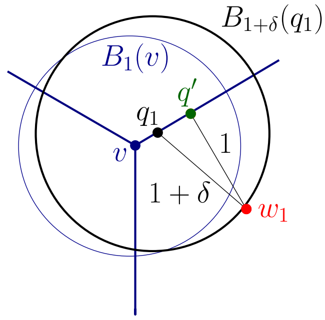


Figure 4 Illustration of the proof of Theorem 2.

For the CLAUSE gadget, we first introduce a NAE-CLAUSE gadget where it is required that not all three values in a clause are equal. For the construction, see Figure 2. Let q_1 be the point on e_1 with distance ε to w_1 and w_2. To force walking back and forth along e_1 for a combination of labels which we want to exclude, we have to ensure that a path from w_1 to w_2 leaves $B_\varepsilon(q_1)$ but stays, once entered, inside $B_\varepsilon(v)$. For the other pairs, (w_1, w_3) and (w_2, w_3) we do the same. A possible drawing of these paths maintaining the planarity of G_2 is shown in the lower sketch of Figure 2. Suppose we map v to $s(v)$ as shown in the Figure 2. Then, edges e_2 and e_3 can be mapped to paths through edges labeled TRUE. But we cannot map e_1 to such a path P: When P reaches vertex w_1, any corresponding reparameterization of e_1 realizing $\delta_F(e_1, P) \leq \varepsilon$ must have reached q_1 as q_1 is the only point with distance at most ε to w_1 on e_2. As P leaves $B_\varepsilon(q_1)$ between w_1 and w_2 and any point on e_1 with distance at most ε to the part of P outside $B_\varepsilon(q_1)$ lies between v and q_1 it follows that $\delta_F(e_1, P) > \varepsilon$. For symmetric reasons it follows that any other all-equal labeling cannot be realized. However, there is a placement of v, such that all three edges e_1, e_2 and e_3 can be mapped to a path in G_2 with Fréchet distance at most ε, for every not-all-equal labeling. Note that MONOTONE-PLANAR-NAE-3-SAT is in P but, as shown in Figure 2, we can use the NAE-CLAUSE gadget as the core of the CLAUSE gadget.

Placing the other gadgets with no overlap (using the WIRE gadget) and noting that all constructed subgraphs are plane, we can, given a MP3S instance A, construct plane graphs G_1 and G_2 such that a map from G_1 to G_2, which realizes $\vec{\delta}_G(G_1, G_2) < \varepsilon$, induces a solution of A: For each positive NAE-clause, at least one of the outgoing edges of G_1 must be mapped to a path through an edge labeled TRUE and thus the corresponding variable v gets the value TRUE. In this case, v cannot set any of the negative clauses TRUE, because the other outgoing edge must be mapped to a path through the edge of G_2 labeled FALSE and this signal can never be switched to TRUE. The same holds for the case of negative NAE-clauses.

Conversely, given a solution S of the MP3S instance A, we can construct a placement of G_1 by choosing p_1 in the VARIABLE gadget for each variable with a TRUE label in S and p_2 for each variable with a FALSE label in S. All edges of the other gadget can be mapped to G_2 in a signal preserving manner. Note that if there exists a clause C in A with three positive labeled variables in S, we change one signal in the PERMUTE gadget from TRUE to FALSE. Thus, we have found a mapping realizing $\vec{\delta}_G(G_1, G_2) \leq \epsilon$. ◀

The characteristics of the gadget still hold for a slightly bigger value of ε which leads to:

▶ **Theorem 2.** *For plane graphs G_1, G_2 it is NP-hard to approximate the directed graph distance $\vec{\delta}_G(G_1, G_2)$ within a 1.10566 factor.*

Proof. We give a detailed proof for the NAE-Clause gadget and note that a similar argument holds for the other gadgets. See Figure 4 for an illustration of the proof.

Let us fix $\varepsilon = 1$. We draw the green spike in the NEA-Clause gadget such that its peak is arbitrarily close to the intersection of a straight line through the edge e_1 and the 1-circle around v. Now, we need to compute the smallest value δ_{min}, such that $B_1(v)$ is completely contained in $B_{1+\delta_{min}}(q_1)$. Then, for any value $\delta < \delta_{min}$, there exists a drawing of the spikes, such that the characteristics of the NAE-Clause gadget still hold, e.g., there is no placement of v allowing an all-equal-labeling.

Note that δ_{min} equals the distance from q_1 to v, when q_1 is at distance $1 + \delta_{min}$ to w_1. Let q' be the position of q_1 for $\delta = 0$ and let d be the distance between q' and q_1. Then we have $\tan(30°) = \frac{\delta_{min}+d}{1} = \delta_{min} + d$. Furthermore, we have $d = \sqrt{(1 + \delta_{min})^2 - 1}$ and therefore $\delta_{min} = \tan(30°) - \sqrt{(1 + \delta_{min})^2 - 1}$, which solves to $\delta_{min} = \frac{1}{4} - \frac{1}{4\sqrt{3}} \approx 0.10566$. The factor by which ε can be multiplied is greater than $1 + \delta_{min}$ for all other gadgets. Thus, δ_{min} is the critical value for the whole construction and the theorem follows. ◀

3 Algorithms for Plane Graphs

Our general algorithm consists of the following four steps. **1.** Compute ε-placements of vertices, **2.** Compute reachability information, **3.** Prune invalid ε-placements, **4.** Based on the remaining ε-placements, decide if there exists a mapping from G_1 to G_2 realizing $\vec{\delta}_{(w)G}(G_1, G_2) \leq \varepsilon$. See [6, 10] for a detailed presentation.

Deciding the Graph Distance in Exponential Time A brute-force method to decide the directed graph distance is to iterate over all possible combinations of valid vertex placements. For each such combination, we iterate over all edges of G_1 to determine whether the vertex placements allow to map each edge to a path with Fréchet distance smaller than ε. This can be done in constant time per edge using the previously computed reachability information. Thus, the runtime is $O(m_1 \cdot m_2^{n_1})$, where $n_i = |V_i|$ and $m_i = |E_i|$.

An alternative approach is the following, which in essence is an extension of step 3 of the general algorithm. First, we remove all tree-like substructures of G_1 and place these as described in [10]. Next, we decompose the remainder of G_1 into chordless cycles, where a chord is a maximal path in G_1 incident to two faces (see Figure 5). We place the parts of G_1 from bottom up, deciding in each step if we can place two adjacent cycles and all the nested substructures of the cycles simultaneously. The time and space complexity of this approach are summarized in the following Theorem. For more details and a proof, see [6].

▶ **Theorem 3.** *For plane graphs, the graph distance can be decided in $O(Fm_2^{2F-1})$ time and $O(m_2^{2F-1})$ space, where F is the number of faces of G_1.*

Note that this method is superior to the brute-force method if $2F - 1 < n_1$.

Polynomial Time Approximation Algorithm The general algorithmic approach yields a good approximation for deciding the graph distance for plane graphs with some geometric restrictions. Again, the decision is based on the existence of valid ε-placements. Thus, the runtime is the same as for the case where G_1 is a tree ($O(n_1 \cdot m_2^2)$ time and space).

▶ **Theorem 4.** *Let $G_1 := (V_1, E_1)$ and $G_2 := (V_2, E_2)$ be plane graphs. Assume that each edge of G_1 has length greater than 2ε. Let α_v be the smallest angle between two edges of G_1 incident to vertex v with $deg(v) \geq 3$, and let $\alpha := \frac{1}{2}\min_{v\in V_1}(\alpha_v)$. If there exists at least one valid ε-placement for each vertex of G_1, then $\vec{\delta}_G(G_1, G_2) \leq \frac{1}{\sin(\alpha)}\varepsilon$.*

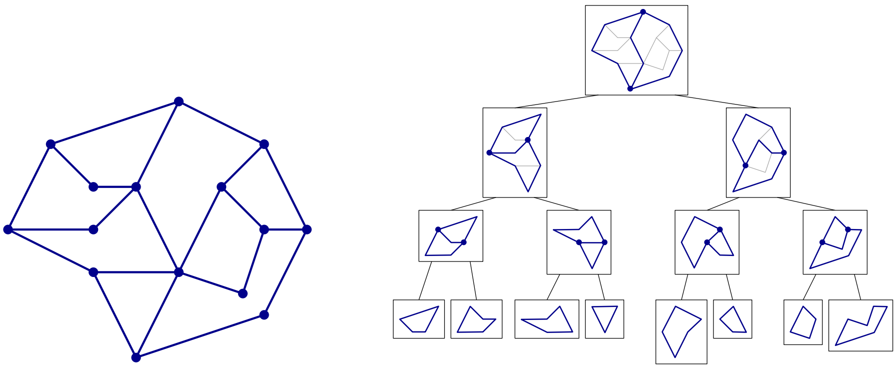


Figure 5 A plane graph is recursively decomposed into chordless cycles.

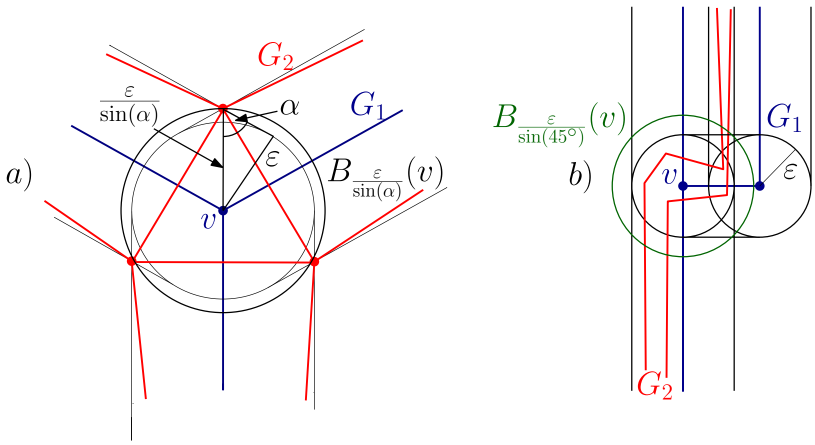


Figure 6 Illustration of the proof of Theorem 4

.

Proof. Let α be the smallest angle between two edges incident to a vertex v with degree at least three and let $C_1, C_2, \ldots, C_j$ be the valid placements of v for a given distance value ε. Furthermore, let V_{C_i} be the set of vertices of C_i. It can be easily shown that for a larger distance value of $\varepsilon_1 \geq \frac{1}{\sin(\alpha)}\varepsilon$ there exist vertices $v_1, v_2, \ldots, v_k$, embedded inside B_{ε_1}, such that the subgraph $C = (V', E')$, where $V' = \bigcup_{i=1}^{j} V_{C_i} \cup \{v_1\} \cup \{v_2\} \cup \cdots \cup \{v_k\}$ and $E' = \{e = \{uw\} \in E_2 | u \in V', w \in V'\}$ is connected (see Figure 6a)). Note that this property is not true if we allow edges with length smaller than 2ε (see Figure 6b)). However, with the condition of a minimal edge length of 2ε, there is only one valid $\frac{1}{\sin(\alpha)}\varepsilon$-placement C for each vertex with degree at least three. Furthermore, every valid ε placement is a valid $\frac{1}{\sin(\alpha)}\varepsilon$-placement. Now, for two paths which start and/or end at a common vertex v, v is mapped to a point on C. This ensures that each edge of G_1 is mapped correctly. ◀

References

1 Mahmuda Ahmed, Brittany T. Fasy, Kyle S. Hickmann, and Carola Wenk. Path-based distance for street map comparison. *ACM Transactions on Spatial Algorithms and Systems*, 28 pages, 2015.

2 Mahmuda Ahmed, Brittany Terese Fasy, and Carola Wenk. Local persistent homology based distance between maps. In *22nd ACM SIGSPATIAL GIS*, pages 43–52, 2014.

3 Mahmuda Ahmed, Sophia Karagiorgou, Dieter Pfoser, and Carola Wenk. A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica*, 19(3):601–632, 2015.

4 Mahmuda Ahmed, Sophia Karagiorgou, Dieter Pfoser, and Carola Wenk. *Map Construction Algorithms*. Springer, 2015.

5 Mahmuda Ahmed and Carola Wenk. Constructing street networks from gps trajectories. In *Proceedings of the 20th Annual European Conference on Algorithms*, ESA'12, pages 60–71, Berlin, Heidelberg, 2012. Springer-Verlag. URL: `http://dx.doi.org/10.1007/978-3-642-33090-2_7`, `doi:10.1007/978-3-642-33090-2_7`.

6 Hugo A. Akitaya, Maike Buchin, Bernhard Kilgus, Stef Sijben, and Carola Wenk. Distance measures for embedded graphs. arXiv e-print, 2018. `https://arxiv.org/abs/1812.09095`.

7 Helmut Alt, Alon Efrat, Günter Rote, and Carola Wenk. Matching planar maps. *Journal of Algorithms*, 49(2):262 – 283, 2003.

8 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(1&2):75–91, 1995.

9 James Biagioni and Jakob Eriksson. Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation Research Record: Journal of the Transportation Research Board*, 2291:61–71, 2012.

10 Maike Buchin, Stef Sijben, and Carola Wenk. Distance measures for embedded graphs. In *Proc. 33rd European Workshop on Computational Geometry (EuroCG)*, pages 37–40, 2017.

11 Otfried Cheong, Joachim Gudmundsson, Hyo-Sil Kim, Daria Schymura, and Fabian Stehn. Measuring the similarity of geometric graphs. In *International Symposium on Experimental Algorithms*, pages 101–112, 2009.

12 Jonathan J. Davies, Alastair R. Beresford, and Andy Hopper. Scalable, distributed, real-time map generation. *IEEE Pervasive Computing*, 5(4):47–54, 2006.

13 Sophia Karagiorgou and Dieter Pfoser. On vehicle tracking data-based road network generation. In *20th ACM SIGSPATIAL GIS*, pages 89–98, 2012.