

Testing Whether a Subgraph Is Convex or Isometric

Sergio Cabello  

Faculty of Mathematics and Physics, University of Ljubljana, Slovenia
Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia

Abstract

We consider the following two algorithmic problems: given a graph G and a subgraph $H \subseteq G$, decide whether H is an isometric or a geodesically convex subgraph of G . It is relatively easy to see that the problems can be solved by computing the distances between all pairs of vertices. We provide a conditional lower bound showing that, for sparse graphs with n vertices and $\Theta(n)$ edges, we cannot expect to solve the problem in $O(n^{2-\varepsilon})$ time for any constant $\varepsilon > 0$. We also show that the problem can be solved in subquadratic time for planar graphs and in near-linear time for graphs of bounded treewidth. Finally, we provide a near-linear time algorithm for the setting where G is a plane graph and H is defined by a few cycles in G .

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases convex subgraph, isometric subgraph, plane graph

Digital Object Identifier 10.4230/LIPIcs.WADS.2025.12

Related Version *Full Version:* <https://arxiv.org/abs/2502.16193>

Funding Funded in part by the Slovenian Research and Innovation Agency (P1-0297, N1-0218, N1-0285). Funded in part by the European Union (ERC, KARST, project number 101071836). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Acknowledgements I am very grateful to Vladimir Batagelj and Sandi Klavžar for discussing the closest related work. I am also very grateful to a reviewer that provided several relevant references.

1 Introduction

Isometric and geodesically convex subgraphs¹ are two natural and fundamental concepts in the area of metric graph theory [3, 39]. Our objective in this paper is to study the algorithmic problem of recognizing whether a given subgraph is convex or isometric.

Setting. Let G be an undirected graph with abstract, positive edge lengths $\ell : E(G) \rightarrow \mathbb{R}_{>0}$. We call G the *host graph* and assume henceforth that it is *connected*. The *length* of a walk (or path) π in G , denoted by $\ell(\pi)$, is the sum of the lengths of its edges; thus $\ell(\pi) = \sum_{e \in E(\pi)} \ell(e)$, where the sum is with multiplicity. For any two vertices $x, y \in V(G)$, a *shortest path* from x to y is a minimum-length path connecting x and y , and the *distance* between x and y in G , denoted by $d_G(x, y)$, is the length of a shortest path from x to y .

¹ There are several distinct concepts of convexity in graphs [11, 39]. Since in this work we only talk about geodesic convexity, we will drop the adjective “geodesically”.



© Sergio Cabello;

licensed under Creative Commons License CC-BY 4.0

19th International Symposium on Algorithms and Data Structures (WADS 2025).

Editors: Pat Morin and Eunjin Oh; Article No. 12; pp. 12:1–12:16

Leibniz International Proceedings in Informatics



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

12:2 Testing Whether a Subgraph Is Convex or Isometric

For any two vertices x, y of G , the *interval* $I_G(x, y)$ is the subgraph of G defined by the union of all shortest paths from x to y . Formally

$$\begin{aligned} V(I_G(x, y)) &= \{u \in V(G) \mid d_G(x, u) + d_G(u, y) = d_G(x, y)\}. \\ E(I_G(x, y)) &= \{uv \in E(G) \mid d_G(x, u) + \ell(uv) + d_G(v, y) = d_G(x, y) \text{ or} \\ &\quad d_G(x, v) + \ell(uv) + d_G(u, y) = d_G(x, y)\}. \end{aligned}$$

A subgraph H of G is (geodesically) *convex* (in G) if and only if

$$\forall x, y \in V(H) : I_G(x, y) \subseteq H,$$

and it is *isometric* (in G) if and only if

$$\forall x, y \in V(H) : d_G(x, y) = d_H(x, y).$$

We consider the following algorithmic problems:

- Given a host graph G and a subgraph $H \subseteq G$, decide whether H is an isometric subgraph.
- Given a host graph G and a subgraph $H \subseteq G$, decide whether H is a convex subgraph.

In graph theory, the concepts of isometric and (geodesically) convex subgraphs are defined for the setting where $\ell(e) = 1$ for all $e \in E(G)$. In this unit-length setting, only induced subgraphs are considered because non-induced graphs cannot be isometric nor convex. In the general case with edge lengths, we may have $d_G(x, y) < \ell(xy)$ for some edges xy of G .

Henceforth, we use n and m for the number of vertices and edges of the host graph G .

Related work. Dourado et al. [16] provide an efficient algorithm to test convexity in the case of unit-length edges. Their approach can be easily adapted to graphs with arbitrary edge lengths, as follows. For each vertex x of the subgraph H , one computes the graph $H_x = \bigcup_{y \in V(H)} I_G(x, y)$. The subgraph H is convex in G if and only if H_x is contained in H for all $x \in V(H)$. The computation of H_x takes $O(m)$ time once we have a shortest-path tree in G from x , and testing whether $H_x \subseteq H$ also takes $O(m)$ time.

We conclude that testing whether a subgraph is convex can be done by computing a shortest-path tree from each vertex, the co-called all-pairs shortest-path (APSP) problem, followed by tests that take $O(nm)$ time. With a simple modification, the same time bound holds for testing whether the given subgraph is isometric. The running time to solve the APSP problem depends on the assumptions about the edge lengths:

- for unit edge lengths, the APSP problem is solved in $O(nm)$ using a simple BFS;
 - when the edge lengths are natural numbers and multiplications take constant time, Thorup [46] solves the APSP problem in $O(nm)$ time;
 - if we only perform additions and comparisons of edge lengths, then the algorithm of Pettie and Ramachandran [40] for APSP takes $O(nm \alpha(m, n))$ time. Using Dijkstra's algorithm with Fibonacci heaps [20], the running time is slightly worse, $O(n^2 \log n + nm)$.
- We conclude that testing whether a subgraph is convex or isometric can be done in $\tilde{O}(nm)$ time in all situations.

Convex subgraphs play an important role in the theory of *median graphs* [12, 35, 36, 37], a class of graphs that has received substantial attention [3, 24, 31]. For median graphs, convex sets and so-called *gated sets* are the same concept; see for example [4, Lemma 3.3]. This can be used to test convexity in linear time: it suffices to check whether every vertex outside H has at most one neighbor in H . Imrich and Klavžar [27] provided a characterization of convex subgraphs in bipartite graphs in terms of Θ -classes. The characterization is for unit

edge lengths and leads to an algorithm taking $O(nm)$ time. They used it to obtain a fast algorithm to recognize median graphs. See also [23]. It was later shown that the recognition of median graphs is closely related to the detection of triangles in graphs [28]. If the host graph G is the Cartesian product of graphs, $G = G_1 \square \dots \square G_k$, and the edges have unit length, a subgraph $H \subseteq G$ is convex if and only if $H = H_1 \square \dots \square H_k$, where each H_i is a convex subgraph of G_i ; see [24, Lemma 6.5]. Together with linear-time algorithms for factorization of graphs [29], this leads to an efficient algorithm for Cartesian products of substantially smaller graphs. Convex subgraphs are being considered also in more applied areas; see for example [21, 34, 44, 45].

A graph is a *partial cube* [15, 19, 48] if it is isomorphic to an isometric subgraph of some hypercube Q_d . Partial cubes enjoy a rich structure and appear in several different combinatorial settings; see for example [3, 24, 38]. Eppstein [18] provided an algorithm to test in $O(n^2)$ time whether a given graph with n vertices is a partial cube. Note that this setting is different to ours because we are not given a subgraph of Q_d , but have to test whether a given graph is isomorphic to an isometric subgraph of Q_d . In our setting, we are given a subgraph. The concept of isometric embeddings in the hypercube or other products of graphs has also been studied recently for weighted graphs; see [5, 42].

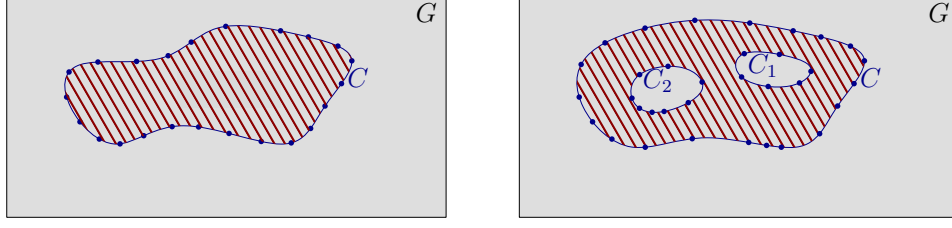
Our contribution. We first provide a series of characterizations of isometric (Section 2) and convex subgraphs (Section 3). Most notably, one of the characterizations uses a marked version of the so-called Wiener index, which we will define below. We think that the characterizations are new and of independent interest. Although proving each of our characterizations is not difficult, identifying them is closely related to the development of our algorithms.

Our first result regarding algorithms is the following conditional lower bound: if G has n vertices and $\Theta(n)$ edges, then there is no algorithm to test whether a given subgraph H of G is isometric or convex in $O(n^{2-\varepsilon})$ time, for any constant $\varepsilon > 0$, unless the Orthogonal Vectors Conjecture (OVC) and the Strong Exponential Time Hypothesis (SETH) fail. This means that for general graphs we cannot expect to test whether a subgraph is convex or isometric in, say, $O(nm^{0.999})$ time, unless some standard assumptions are wrong. The lower bound holds for graphs with unit edge lengths and implies that the algorithms mentioned above, with running times $\tilde{O}(nm)$, are asymptotically optimal up to polynomial factors $\Theta(n^\varepsilon)$ for any $\varepsilon > 0$. Our reduction is via the diameter problem. We review OVC and SETH in Section 4, where we also provide our reduction via the diameter problem.

The characterization using the marked version of the Wiener index can be combined with adaptations of known algorithms to obtain the following results about testing whether a given subgraph is isometric or convex:

- If the host graph is planar graph, the test can be performed in $\tilde{O}(n^{5/3})$ time.
- If the host graph has treewidth bounded by a constant $k \geq 3$, the test can be performed in $O(n \log^{k-1} n)$ time; the constant in the O -notation depends on k . When the treewidth k is considered a parameter, the test can be performed in $n^{1+\varepsilon} 2^{O(k)}$, for any constant $\varepsilon > 0$; the constant in the O -notation depends on the choice of ε .

Finally, in Section 6 we consider the following setting. The host graph G is a *plane graph*, that is, a planar graph with a fixed embedding in the plane. Let C be a cycle in G and let $H = \text{int}(G, C)$ be the subgraph of G bounded by C and including C ; see Figure 1 left. We show how to decide whether H is an isometric or a convex subgraph of G in $O(n \log n)$ time. The result can be extended to subgraphs of G defined by a constant number of cycles; see the right of Figure 1 for an example and Section 6 for more precise definitions. Compared



■ **Figure 1** Schematic view of the subgraph $H = \text{int}(G, C)$ defined by the cycle C (left) and the subgraph $\text{holes}(G, C, \{C_1, C_2\})$ defined by a few cycles (right). They are both marked with stripes.

to the other results presented in the paper, this result is interesting because it uses different characterizations and a different algorithmic paradigm. We use multiple-source shortest-path (MSSP) algorithms for planar graphs [9, 32] and combine the distances in the interior and the exterior of C , as the source of the shortest-path tree moves along the cycle C . As far as we know, this is the first application of MSSP where we maintain a combination of the interior and the exterior distances.

Notation and basic concepts. Through our discussion we will assume that the host graph G is fixed and in the notation we will often drop the dependency on G .

For each subgraph H of G , the **boundary** of H (in G) is the set ∂H of vertices of H that are incident to some edge outside H . Thus, $\partial H = \{v \in V(H) \mid \exists uv \in E(G) \setminus E(H)\}$. Note that each path in G that has edges from $E(H)$ and edges from $E(G) \setminus E(H)$ must have some vertex of ∂H .

For each subgraph H of G , let H_{out} be the **complement** of H in G , defined as the subgraph with edges not contained in H . Thus, $H_{\text{out}} = (V(G), E(G) \setminus E(H))$. The graph H_{out} may have isolated vertices. For distances in H_{out} we write $d_{H_{\text{out}}}(\cdot, \cdot)$.

We will often use without explicit mention that for any subgraph H of G and any vertices $x, y \in V(H)$ we have $d_G(x, y) \leq d_H(x, y)$ and $d_G(x, y) \leq d_{H_{\text{out}}}(x, y)$.

The **Wiener index with marked vertices** for a graph G with respect to $U \subseteq V(G)$ is $\sum_{x, y \in U} d_G(x, y)$. The classical Wiener index is the case of $U = V(G)$. For unweighted graphs, this is a very similar to the vertex-weighted medians considered by Bandelt and Chepoi [2] and it is a particular case of the vertex-weighted Wiener index considered by Bénéteau et al. [4] for median graphs. In their weighted version, the marked vertices have weight 1 and the vertices to be ignored in the sum have weight 0.

2 Characterizing isometric subgraphs

In this section we provide different criteria to test when a subgraph H of G is isometric.

► **Lemma 1.** *Let G be a connected graph and let H be a subgraph of G . Then H is an isometric subgraph of G if and only if*

$$\forall x, y \in \partial H : d_H(x, y) \leq d_{H_{\text{out}}}(x, y).$$

The next two characterizations use the Wiener index with marked vertices.

► **Lemma 2.** *Let G be a connected graph and let H be a subgraph of G . Then H is an isometric subgraph of G if and only if*

$$\sum_{x, y \in V(H)} d_G(x, y) = \sum_{x, y \in V(H)} d_H(x, y).$$

In fact, it suffices to add the distances between vertices on the boundary of H , or any superset of them.

► **Lemma 3.** *Let G be a connected graph, let H be a subgraph of G and let U be such that $\partial H \subseteq U \subseteq V(H)$. Then H is an isometric subgraph of G if and only if*

$$\sum_{x,y \in U} d_H(x,y) = \sum_{x,y \in U} d_G(x,y).$$

3 Characterizing convex subgraphs

Here we provide criteria to identify when a subgraph H of G is convex. The structure of the statements and the proofs is parallel to the isometric case considered in Section 2.

► **Lemma 4.** *Let G be a connected graph and let H be a subgraph of G . Then H is a convex subgraph of G if and only if*

$$\forall x, y \in \partial H, x \neq y: d_H(x, y) < d_{H_{\text{out}}}(x, y).$$

For the following characterizations, we are going to decrease slightly the length of some edges of G so that no new shortest paths are created. For this, we define

$$\varepsilon = \varepsilon(G, \ell) = \frac{\min\{\ell(\pi) - \ell(\pi') \mid \pi, \pi' \text{ paths in } G \text{ with } \ell(\pi) > \ell(\pi')\}}{n}.$$

First note that we can indeed take the minimum because there is a finite number of paths in G . Moreover, for each edge $e \in E(G)$, we have $\varepsilon \leq \ell(e)/n$ because e is a path in G , as well as the empty path. For a subgraph H of G and a δ with $0 < \delta \leq \varepsilon$, we define the edge lengths $\hat{\ell}$ by shortening the edges *not* in H by δ . Thus

$$\hat{\ell}(e) = \begin{cases} \ell(e) & \text{if } e \in E(H), \\ \ell(e) - \delta & \text{if } e \in E(G) \setminus E(H). \end{cases}$$

Note that $\hat{\ell}(e) > 0$ for all $e \in E(G)$ because $\delta \leq \varepsilon < \ell(e)$. The lengths $\hat{\ell}$ depend on H and δ , but we drop this dependency to avoid cluttering the notation. Let $\hat{d}_G(\cdot, \cdot)$ denote the distance in G with respect to the edge lengths $\hat{\ell}$.

► **Lemma 5.** *If G is connected and $0 < \delta \leq \varepsilon$, each shortest path in G with respect to $\hat{\ell}$ is a shortest path in G with respect to ℓ . (The converse is not necessarily true.)*

► **Lemma 6.** *Let G be a connected graph, let H be a subgraph of G , let δ be a real value such that $0 < \delta \leq \varepsilon$, and let \hat{d}_G be the corresponding perturbed distance. Then H is a convex subgraph of G if and only if*

$$\sum_{x,y \in V(H)} \hat{d}_G(x,y) = \sum_{x,y \in V(H)} d_H(x,y).$$

Proof. Assume first that H is a convex subgraph of G . Consider any two vertices x, y of H and let $\pi_{x,y}$ be a shortest path in G from x to y with respect to $\hat{\ell}$. Therefore $\hat{d}_G(x, y) = \hat{\ell}(\pi_{x,y})$. Because of Lemma 5, the path $\pi_{x,y}$ is a shortest path in G with respect to ℓ , which means that $\ell(\pi_{x,y}) = d_G(x, y)$. Because H is convex, the path $\pi_{x,y}$ is contained in H , and since the edges of H have the same length in ℓ and in $\hat{\ell}$, we have $\hat{\ell}(\pi_{x,y}) = \ell(\pi_{x,y})$. Putting the forthcoming equalities together and using that H is a convex (and thus isometric) subgraph of G , we obtain

$$\hat{d}_G(x, y) = \hat{\ell}(\pi_{x,y}) = \ell(\pi_{x,y}) = d_G(x, y) = d_H(x, y).$$

12:6 Testing Whether a Subgraph Is Convex or Isometric

Since this equality holds for each $x, y \in V(H)$, we obtain

$$\sum_{x,y \in V(H)} \hat{d}_G(x, y) = \sum_{x,y \in V(H)} d_H(x, y).$$

To show the other direction, note first that $\hat{d}_G(x, y) \leq d_G(x, y) \leq d_H(x, y)$ for all $x, y \in V(H)$ because $\hat{\ell}(e) \leq \ell(e)$ for all edges $e \in E(G)$ (for the first inequality) and H is a subgraph of G (for the second inequality). Assume now that H is not a convex subgraph of G . This means that there exists vertices $x_0, y_0 \in V(H)$ and a x_0 -to- y_0 shortest path π_0 that is not contained in H . Since π_0 has some edge in $E(G) \setminus E(H)$, we have $\hat{\ell}(\pi_0) < \ell(\pi_0)$ and therefore $\hat{d}_G(x_0, y_0) < d_G(x_0, y_0) \leq d_H(x_0, y_0)$. Therefore

$$\begin{aligned} \sum_{x,y \in V(H)} \hat{d}_G(x, y) &= 2 \cdot \hat{d}_G(x_0, y_0) + \sum_{\substack{x,y \in V(H) \\ \{x,y\} \neq \{x_0,y_0\}}} \hat{d}_G(x, y) \\ &< 2 \cdot d_H(x_0, y_0) + \sum_{\substack{x,y \in V(H) \\ \{x,y\} \neq \{x_0,y_0\}}} d_H(x, y) = \sum_{x,y \in V(H)} d_H(x, y). \quad \blacktriangleleft \end{aligned}$$

► **Lemma 7.** *Let G be a connected graph, let H be a subgraph of G , let U be such that $\partial H \subseteq U \subseteq V(H)$, let δ be a real value such that $0 < \delta \leq \varepsilon$, and let \hat{d}_G be the corresponding perturbed distance. Then H is a convex subgraph of G if and only if*

$$\sum_{x,y \in U} d_H(x, y) = \sum_{x,y \in U} \hat{d}_G(x, y).$$

Proof. In the first part of the proof of Lemma 6 we have seen that, if H is a convex subgraph of G , then for all $x, y \in V(H)$ we have $\hat{d}_G(x, y) = d_H(x, y)$. The equality of the two sums follows.

For the other direction, we start assuming that H is not a convex subgraph of G . If H is not connected, then the left sum is ∞ while the right side is bounded, and therefore they are distinct. We continue assuming that H is connected. By Lemma 4, there exist distinct vertices $x_0, y_0 \in \partial H$ such that $d_H(x_0, y_0) \geq d_{H_{\text{out}}}(x_0, y_0)$. Let π_0 be a shortest x_0 -to- y_0 path in H and let π'_0 be a shortest x_0 -to- y_0 path in H_{out} . Note that π'_0 indeed exists because $d_{H_{\text{out}}}(x_0, y_0)$ is bounded. We then have $\ell(\pi_0) = d_H(x_0, y_0) \geq d_{H_{\text{out}}}(x_0, y_0) = \ell(\pi'_0)$. Since π'_0 has edges outside $E(H)$, we have $\hat{\ell}(\pi'_0) < \ell(\pi'_0)$, and therefore

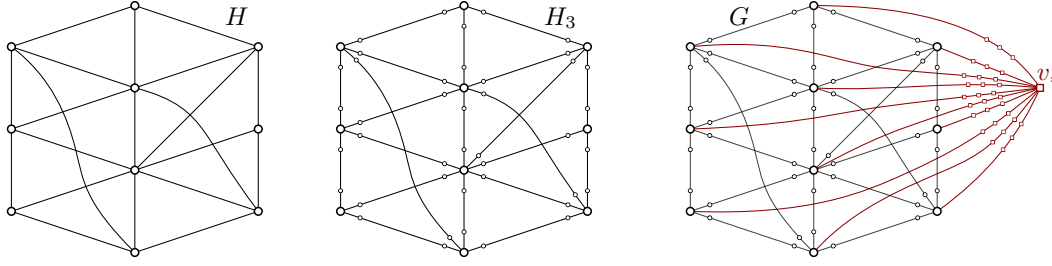
$$\hat{d}_G(x_0, y_0) \leq \hat{\ell}(\pi'_0) < \ell(\pi'_0) \leq \ell(\pi_0) = d_H(x_0, y_0).$$

In summary, $\hat{d}_G(x_0, y_0) < d_H(x_0, y_0)$ for some $x_0, y_0 \in \partial H$. In general, for all $x, y \in V(H)$ we have $\hat{d}_G(x, y) \leq d_G(x, y) \leq d_H(x, y)$ because $\hat{\ell}(e) \leq \ell(e)$ for all edges $e \in E(G)$ (for the first inequality) and H is a subgraph of G (for the second inequality). From this it follows that for U such that $\partial H \subseteq U \subseteq V(H)$ we have

$$\sum_{x,y \in U} \hat{d}_G(x, y) < \sum_{x,y \in U} d_H(x, y). \quad \blacktriangleleft$$

4 Conditional lower bound

Our conditional lower bound is based on the following assumptions, where $\langle \cdot, \cdot \rangle$ is the inner product and the diameter of a graph G is $\text{diam}(G) = \max_{x,y \in V(G)} d_G(x, y)$.



■ **Figure 2** Example for the construction of Theorem 9. Left: a graph H . Center: the graph H_3 . Right: the graph G with H_3 as a subgraph.

► **Orthogonal Vectors Conjecture (OVC, [47]).** *For every $\varepsilon > 0$ there is a $c \geq 1$ such that the following problem cannot be solved in $O(n^{2-\varepsilon})$ time: given n binary vectors $v_1, \dots, v_n \in \{0, 1\}^d$, where $d = \lceil c \log n \rceil$, are there $i, j \in [n]$ such $\langle v_i, v_j \rangle = 0$?*

► **Strong Exponential Time Hypothesis (SETH, [25, 26]).** *For every $\varepsilon > 0$ there exists an integer $k \geq 3$ such that k -SAT with n variables cannot be solved in $O(2^{(1-\varepsilon)n})$ time.*

Williams [47] posed OVC and showed that it is implied by SETH. One of the interesting consequences of OVC, and therefore SETH, is the following result.

► **Theorem 8** (Consequence of Theorem 9 in Roditty and Vassilevska Williams [41]). *If one can distinguish in $O(n^{2-\varepsilon})$ time for some constant $\varepsilon > 0$ between diameter 2 and 3 in undirected graphs with n vertices and $O(n)$ edges, all of unit length, then OVC and SETH fail.*

► **Theorem 9.** *Assume that, for some $\varepsilon > 0$, there is an algorithm that in $O(n^{2-\varepsilon})$ time solves the following problem: given an undirected graph G with n vertices and $O(n)$ edges, all with unit length, and given a subgraph H of G , decide whether H is an isometric subgraph of G . Then OVC and SETH fail.*

The same result holds for deciding if H is a convex subgraph of G .

Proof. We will reduce from the diameter problem considered in Theorem 8. Let H be a graph with $O(n)$ vertices and $O(n)$ edges, all edges of unit length, for which we want to distinguish whether it has diameter 2 or 3. Consider the graph $G = G(H)$ defined as follows; see Figure 2 for an example. We first construct the graph H_3 from H by subdividing each edge of H twice. Then, we add a new vertex v_* and connect it to each vertex of $V(H)$ with a 4-edge path using new interior vertices. Let $G = G(H)$ be the resulting graph. Note that H_3 is a subgraph of G and $\partial H_3 = V(H)$.

We assign unit length to each edge of G . For all $x, y \in V(H)$, we have $d_{H_3}(x, y) = 3 \cdot d_H(x, y)$ and $d_G(x, y) = \min\{3 \cdot d_H(x, y), 8\}$, where the 8 comes from the path in G through v_* .

We now show that H_3 is an isometric subgraph of G if and only if H has diameter 2. If H has diameter 2, then for all $x, y \in V(H)$ we have

$$d_G(x, y) = \min\{3 \cdot d_H(x, y), 8\} = 3 \cdot d_H(x, y) = d_{H_3}(x, y).$$

Since $\partial H_3 = V(H)$, we have $\sum_{x, y \in \partial H_3} d_G(x, y) = \sum_{x, y \in \partial H_3} d_{H_3}(x, y)$ and H_3 is an isometric subgraph of G because of Lemma 3. If H has diameter 3, then there is some $x_0, y_0 \in V(H)$ with $d_H(x_0, y_0) = 3$, and therefore $d_{H_3}(x_0, y_0) = 3 \cdot d_H(x_0, y_0) = 9 > 8 \geq d_G(x_0, y_0)$. It follows that H_3 is not an isometric subgraph of G .

We now show that H_3 is a convex subgraph of G if and only if H has diameter 2. Let \hat{d} denote the distance in G when we decrease the length of the edges of $E(G) \setminus E(H_3)$ by a sufficiently small value $\delta > 0$. For concreteness, we can take $\delta < 1/10$ in our discussion. Then, for all $x, y \in V(H)$, we have $d_{H_3}(x, y) = 3 \cdot d_H(x, y)$ and $\hat{d}_G(x, y) = \min\{3 \cdot d_H(x, y), 8 - 8\delta\}$.

If H has diameter 2, then, for all $x, y \in V(H)$, we have

$$\hat{d}_G(x, y) = \min\{3 \cdot d_H(x, y), 8 - 8\delta\} = 3 \cdot d_H(x, y) = d_{H_3}(x, y),$$

where we have used that $3 \cdot d_H(x, y) \leq 6$ and $8\delta < 1$. Since $\partial H_3 = V(H)$, we have $\sum_{x, y \in \partial H_3} \hat{d}_G(x, y) = \sum_{x, y \in \partial H_3} d_{H_3}(x, y)$, and H_3 is a convex subgraph of G because of Lemma 7.

If H has diameter 3, then we have already seen that H_3 is not an isometric subgraph of G , and thus cannot be a convex subgraph.

If H has $O(n)$ vertices and $O(n)$ edges, then G has $|V(H)| + 2 \cdot |E(H)| + 3 \cdot |V(H)| + 1 = O(n)$ vertices and $3 \cdot |E(H)| + 4 \cdot |V(H)| = O(n)$ edges. The construction of G and H_3 from H takes linear time. It follows that, if we could decide whether H_3 is isometric (or convex) in $O(n^{2-\varepsilon})$ time, for some $\varepsilon > 0$, we could decide whether H has diameter 2 or 3 in $O(n^{2-\varepsilon})$ time, and therefore OVC and SETH would be false by Theorem 8. ◀

5 Subquadratic algorithms for special classes of graphs

In this section we provide algorithms to test whether a subgraph is isometric or convex, when the host graph is planar or has small treewidth. The idea is to use the criteria of Lemmas 2 and 6 together with modifications of previous algorithms to compute the classical Wiener index.

5.1 Shortening some edges a bit

To test whether $H \subseteq G$ is convex using Lemma 6 we have to shorten the edges of $E(G) \setminus E(H)$ by a small enough amount δ , where $0 < \delta \leq \varepsilon(G, \ell)$. We do not know how to compute $\varepsilon(G, \ell)$ efficiently, or some other value that could replace it. We solve this by perturbing the edge lengths infinitesimally, as follows. (For unit length edges this is easy by taking $\varepsilon = 1/2n$.)

We define a new, composite length ℓ_2 for G using 2-tuples. The intuition is that a pair $(a, b) \in \mathbb{R}_{\geq 0} \times \mathbb{Z}$ represents the real value $a + b \cdot \delta$ for an infinitesimal $\delta > 0$. This means that we can compare two lengths (a, b) and (a', b') using the lexicographic comparison: (a, b) is smaller than (a', b') , denoted by $(a, b) \prec \ell(a', b')$, if and only if $a < a'$ or if $a = a'$ and $b < b'$. This lexicographic comparison is compatible with the test of whether $a + b \cdot \delta < a' + b' \cdot \delta$ for an infinitesimal $\delta > 0$.

To carry the perturbation, we define a new edge length ℓ_2 in the following way:

$$\forall e \in E(G) : \quad \ell_2(e) = \begin{cases} (\ell(e), 0) & \text{if } e \in E(H), \\ (\ell(e), -1) & \text{if } e \in E(G) \setminus E(H). \end{cases}$$

The length of a walk (or path) π is defined as the vector-sum of the the 2-tuples $\ell_2(e)$ over $e \in E(\pi)$, with multiplicity. Therefore, we have

$$\ell_2(\pi) = (\ell(\pi), -|E(\pi) \setminus E(H)|),$$

which represents the value $\ell(\pi) - \delta \cdot |E(\pi) \setminus E(H)|$ for an infinitesimal $\delta > 0$.

Classical algorithms for computing shortest paths can compute distances using this new edge lengths. In general, any algorithm that manipulates the edge lengths with additions and makes comparisons of the lengths of paths can be adapted to this 2-tuple distances without

an asymptotic increase in the running time. (For the sake of comparison, it is not clear how to adapt efficiently an algorithm that would multiply edge lengths because powers of δ would start showing up and a constant-size tuple would not suffice anymore.)

5.2 Graphs of small treewidth

Let G be a graph of treewidth $\text{tw}(G)$, possibly with edge lengths, and let H be a subgraph of G . It is well-known that H has treewidth at most $\text{tw}(G)$.

Cabello and Knauer [10] showed that, for each constant $k \geq 3$, the Wiener index of graphs of treewidth at most k can be computed in $O(n \log^{k-1} n)$ time; the constant hidden in the O -notation depends on k . Bringmann, Husfeldt and Magnusson [6] carried out the analysis making the dependency on the treewidth explicit and showed that the Wiener index can be computed in $n^{1+\varepsilon} 2^{O(\text{tw}(G))}$, for each constant $\varepsilon > 0$; the constant hidden in the O -notation depends on ε .

Both papers use data structures for orthogonal range searching. The vertices of G are used to define a point set and to define ranges. It is easy to see that those results can be adapted to compute the sum $\sum_{x,y \in U} d_G(x,y)$ for any given $U \subseteq V(G)$. Indeed, it suffices to define the points and the ranges only for the vertices of U . This is explicitly mentioned in [8, Section 5]. The running times of the results are not affected by this. (Actually, if $\log |U| = o(\log n)$, the asymptotic running times can be slightly improved, but we omit this in our worst-case analysis.)

Next, we note that the algorithms in [6, 10] are manipulating and comparing sums of the edge lengths; they do not make other operations with the edge lengths. This means that they can handle edge lengths defined by a 2-tuple, as discussed in Section 5.1. We conclude that those algorithms can be used, without changing the asymptotic running time, to compute

$$\sum_{x,y \in V(H)} d_G(x,y), \quad \sum_{x,y \in V(H)} \hat{d}_G(x,y), \quad \sum_{x,y \in V(H)} d_H(x,y),$$

where $\hat{d}_G(x,y)$ is defined via 2-tuples to model a shortening of the edges of $E(G) \setminus E(H)$ by an infinitesimal $\delta > 0$. We can then compare them, taking into account that the ℓ_2 -length (a,b) is equal to the ℓ -length a if and only if $b = 0$. We have shown the following.

► **Theorem 10.** *Let $k \geq 3$ be a constant. Given a graph G with n vertices and treewidth at most k , we can test whether a given subgraph of G is convex or isometric in $O(n \log^{k-1} n)$ time. The constant hidden in the O -notation depends on k .*

We can decide whether a given subgraph of G is convex or isometric in $n^{1+\varepsilon} 2^{O(\text{tw}(G))}$ time, for each constant $\varepsilon > 0$. The constant hidden in the O -notation depends on ε .

5.3 Planar graphs

Cabello [7] gave a randomized algorithm to compute the Wiener index of a planar graph G in $\tilde{O}(n^{11/6})$ expected time. This was improved by Gawrychowski et al. [22], who provided a deterministic algorithm taking $\tilde{O}(n^{5/3})$ time. Both algorithms have the property that they can compute $\sum_{x,y \in U} d_G(x,y)$ for any given $U \subseteq V(G)$. Also, both algorithms only use the edge lengths via sums and comparisons of values. Therefore, they can also be used to obtain

$$\sum_{x,y \in V(H)} d_G(x,y), \quad \sum_{x,y \in V(H)} \hat{d}_G(x,y), \quad \sum_{x,y \in V(H)} d_H(x,y),$$

where $\hat{d}_G(x,y)$ is defined via 2-tuples to model a shortening by an infinitesimal $\delta > 0$. This shows the following.

► **Theorem 11.** *Given a planar graph G with n vertices, we can test whether a given subgraph of G is convex or isometric in $\tilde{O}(n^{5/3})$ time.*

5.4 Discussion

We next present classes of graphs illustrating two limitations of our technique based on the Wiener index with marked vertices. One limitation is that for testing convexity we have to consider edge-weights, even if the original graph is unweighted; the other limitation is that subgraphs may lose some useful properties enjoyed by the host graph.

Let us consider the class of K_h -minor-free graphs, where K_h denotes the complete graph on h vertices. Ducoffe et al. [17] showed how to compute the diameter of such graphs in subquadratic time for the unweighted version, but left open the question of computing the Wiener index in subquadratic time. Recent works by Le, Wulff-Nilsen, Karczmarz and Zheng [30, 33] have improved the dependency on h and their new techniques also work for the Wiener index, but still only for the case of unweighted graphs. The currently best result [30, Theorem 1.3] computes the Wiener index in *unweighted* graphs without a K_h -minor in $\tilde{O}(n^{2-1/(3h-2)})$ time. The technique can be easily adapted to compute the Wiener index with marked vertices. Since a subgraph of a K_h -minor-free graph is obviously also K_h -minor-free, we can use this algorithm to test whether a given subgraph is isometric via Lemma 2. We summarize.

► **Theorem 12.** *Given an *unweighted* K_h -minor-free graph G with n vertices, we can test whether a given subgraph of G is isometric in $\tilde{O}(n^{2-1/(3h-2)})$ time.*

Since currently we do not know how to compute in subquadratic time the Wiener index of K_h -minor-free graph, we cannot make the modification of edge-lengths needed to test whether a subgraph is convex via Lemma 6. Therefore, we cannot make an analogous claim regarding testing convexity, even for unweighted graphs.

Let us turn our attention now to unweighted median graphs. Bénéteau et al. [4] show that the Wiener index of median graphs can be computed in linear time. In fact, they solve a vertex-weighted variant that generalizes the Wiener index with marked vertices. Consider the problem of testing whether a given subgraph H of a median graph G is isometric. We can apply their algorithm to compute $\sum_{x,y \in V(H)} d_G(x,y)$ in linear time. However, it is not clear (at least to the author) how to compute in subquadratic time $\sum_{x,y \in V(H)} d_H(x,y)$, even if we assume that H is an isometric subgraph of G , and thus a partial cube. When we consider a subgraph of a median graph, we lose many of its structural properties, even if the subgraph is isometric. Similarly, for testing convexity using the Wiener index, we would need to consider the perturbed weights, which in general does not go well with many of the structural properties of median graphs. (In contrast, as mentioned in the introduction, testing whether a subgraph of a median graph is convex amounts to testing whether it is gated, a property that can be tested in linear time.)

6 Subgraphs of plane graphs defined by cycles

Let G be a plane graph, that is, a planar graph with a fixed embedding in the plane. For each cycle C of G , let γ_C be the Jordan curve defined by C . Such a cycle C defines naturally the *interior graph*, denoted by $\text{int}(G, C)$, which is the subgraph of G contained in the closure of the bounded set of $\mathbb{R}^2 \setminus \gamma_C$. Thus, C is contained in $\text{int}(G, C)$. In this section we consider the problem of testing whether $H = \text{int}(G, C)$ is a convex or isometric subgraph of G . In H we assume the embedding inherited from G and then C is a facial walk of H .

6.1 Data structure for sequences of numbers

We will use a data structure to store a sequence a_1, \dots, a_k of k values in \mathbb{R} , where each value is initialized arbitrarily, and supporting the following operations:

- $\text{INIT}(a_1^0, \dots, a_k^0)$: initializes the data structure, setting $a_i = a_i^0$ for each $i \in [k]$.
- $\text{MIN}()$ returns $\min\{a_1, \dots, a_k\}$.
- $\text{ADD}(i, j, \Delta)$ adds the value $\Delta \in \mathbb{R}$ to a_i, \dots, a_j , where $i \leq j$.

An efficient approach is to use a *static* balanced binary search tree storing k elements with keys $[k] = \{1, \dots, k\}$, such that the node with key i stores a_i . The tree is then augmented with data at the nodes to represent values that have to be added to the whole subtree. See for example Choi, Cabello and Ahn [13] for a comprehensive description. Another alternative, which is a bit overkilling, is to use dynamic trees, such as cut-link trees [43] or top trees [1]; in this case we maintain a path on k nodes, where the i th node stores the value a_i . These data structures allow to add values and to query for the minimum value in a subpath, which in our case corresponds to a contiguous subsequence. We summarize for later reference.

► **Lemma 13.** *There is a data structure to maintain a sequence a_1, \dots, a_k of k numbers that supports the operations MIN , and ADD in $O(\log k)$ time per operation. The initialization of the data structure, INIT , takes $O(k)$ time.*

6.2 Encoding distances within a face

Let C be a facial cycle in a plane graph G with n vertices, and let x_1, \dots, x_k be the vertices as we walk along C , starting from an arbitrary vertex x_1 of $V(C)$. For each $i \in [k]$, let D_i be the sequence of numbers telling the distances from x_i to all other vertices; thus $D_i = (d_G(x_i, x_j))_{j \in [k]}$.

In a seminal work, Klein [32] showed that in $O(n \log n)$ time one can maintain a shortest-path tree of G rooted at a vertex of C as the root moves along C . An alternative point of view based on parametric shortest-path trees was given by Cabello, Chambers and Erickson [9], and a new approach based on divide-and-conquer has been given by Das et al. [14].

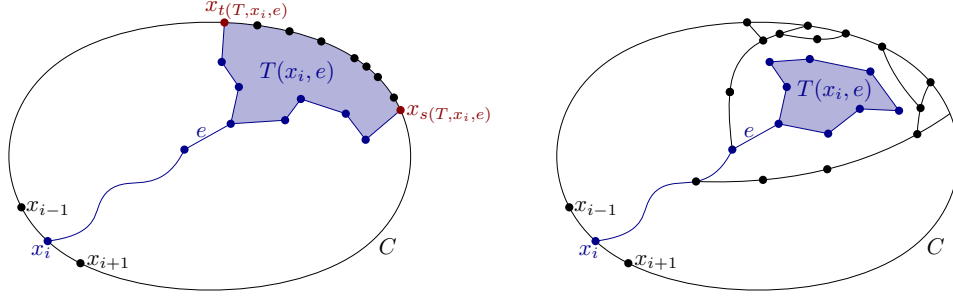
The key insight is that when the root of the shortest path tree moves along the boundary of a face, each directed arc of G appears in the shortest path tree along a contiguous subsequence of roots along the facial walk. In other words, with the move of the root through the facial walk, each directed edge enters and exits the shortest path tree at most once.

These results are the basis for the following property.

► **Lemma 14.** *In $O(n \log n)$ time we can compute sequences $\Lambda_2, \dots, \Lambda_k$ of operations with the following properties*

- for $i = 2, \dots, k$, each Λ_i is a sequence of $|\Lambda_i|$ operations of the type ADD in a sequence a_1, \dots, a_k of numbers;
- for $i = 2, \dots, k$, the sequence D_i of distances from x_i is obtained from the sequence D_{i-1} performing the operations given in Λ_i ;
- the sequences Λ_i together have a linear number of operations; that is $\sum_i |\Lambda_i| = O(n)$.

Proof. Let T be an arbitrary spanning tree of G rooted at a vertex r , let e be an edge of T , and let $T(r, e)$ be the subtree of $T - e$ that does not contain the root r . Because G is plane and C defines a face of G , the vertices of $V(C)$ that appear in $T(r, e)$ form a continuous subsequence of C . See Figure 3. Note that the subsequence may be empty, meaning that $V(C) \cap V(T(r, e))$ is empty. Let $s(T, r, e)$ and $t(T, r, e)$ be the start and the end of the subsequence, if it is non-empty, such that the vertices $\{x_{s(T, r, e)}, x_{s(T, r, e)+1}, \dots, x_{t(T, r, e)}\}$ (indices modulo k) are $V(C) \cap V(T(r, e))$.



■ **Figure 3** Schematic view of the interaction of $T(x_i, e)$ and the face defined by C . On the left, $V(T(x_i, e)) \cap V(C)$ is not empty, but on the right, $V(T(x_i, e)) \cap V(C)$ is empty.

Using data structures for dynamic forests, one can maintain the tree T under swapping of edges (removing an edge and inserting an edge while maintaining a tree) and changing the root such that: each update takes $O(\log n)$ time; the indices $s(T, x_i, e)$ and $t(T, x_i, e)$ can be obtained in $O(\log n)$ time for any query $(x_i, e) \in V(C) \times E(T)$. The basic idea for this is to maintain the spanning tree T^* of the dual graph that uses the edges dual to those in $E(G) \setminus E(T)$. The two edges defining $s(T, x_i, e)$ and $t(T, x_i, e)$ are, in the dual tree, the only edges dual to the edges of $E(C)$ that lie on the unique path of T^* that connects the two faces bounded by e . It is easy to extend data structures for dynamic trees [1, 43] to handle such type of queries. The edges dual to $E(C)$ are marked as special in T^* and we have to find the special edges that lie in a path of T^* between two given nodes. See [9, 32] for data structures with similar properties.

Let T be a tree, let T' be the tree obtained from T by deleting the edge e and inserting the edge e' , let r be a vertex of T and T' , let $A = (a_i = d_T(r, x_i))_{i \in [k]}$ the sequence of distances in T from r to x_i (for $i \in [k]$), and let $A' = (a'_i = d_{T'}(r, x_i))_{i \in [k]}$ the sequence of distances in T' from r to x_i (for $i \in [k]$). Assume that e in T and e' in T' have a common target; this means that in T we have $e = x \rightarrow z$ and in T' we have $e' = y \rightarrow z$, where the vertex z is the same. Then, we can go from A to A' by performing $O(1)$ operations ADD. More precisely, for each vertex x_i that lies in $T(r, e)$ we have to decrease the value a_i by $d_T(r, z)$ and increase it by $d_{T'}(r, z)$. (Because e and e' have the same target we have $V(T(r, e)) = V(T'(r, e'))$.) Each one of these operations is done with at most two operations ADD. For example, if $s(T, r, e) < t(T, r, e)$, then we perform $\text{ADD}(s(T, r, e), t(T, r, e), -d_T(r, z))$, but if $s(T, r, e) > t(T, r, e)$, we split it into the two updates $\text{ADD}(s(T, r, e), k, -d_T(r, z))$ and $\text{ADD}(1, t(T, r, e), -d_T(r, z))$.

Finally, we note that [9, 32] show how to maintain a shortest-path tree as the root moves along the boundary of the face defined by C performing $O(n)$ swaps of the type $T' = (T - e) + e'$ that have the property that e and e' have the same target vertex. When sliding the root from x_i to x_{i+1} , one has to also update the distances because of the change of root. This amounts to adding $\ell(x_i x_{i+1})$ to all the values of $V(C) \cap V(T(x_{i+1}, x_i x_{i+1}))$ and subtracting $\ell(x_i x_{i+1})$ to all the values of $V(C) \cap V(T(x_i, x_i x_{i+1}))$. The result follows. ◀

6.3 Testing

We discuss now how to test if $H = \text{int}(G, C)$ is an isometric subgraph of G . Recall that $E(H_{\text{out}}) = E(G) \setminus E(H)$. We add to H_{out} the edges of C with very large weights, say $2 \sum_{e \in E(G)} \ell(e)$, so that the new edges never appear in a shortest path.

Let x_1, \dots, x_k be the vertices of C ordered as they appear in C . The cycle C defines a face in H and in H_{out} and therefore we can use Lemma 14 for each of those graphs. For H we obtain the sequence of operations $\Lambda_2, \dots, \Lambda_k$ that are needed to maintain

$$D_i = \left(a_i = d_H(x_i, x_j) \right)_{j \in [k]},$$

while for H_{out} we obtain the the sequence of operations $\Lambda'_2, \dots, \Lambda'_k$ that are needed to maintain

$$D'_i = \left(a'_i = d_{H_{\text{out}}}(x_i, x_j) \right)_{j \in [k]}.$$

The objective now is to maintain the sequence of values

$$\tilde{D}_i = \left(\tilde{a}_i = d_{H_{\text{out}}}(x_i, x_j) - d_H(x_i, x_j) = a'_i - a_i \right)_{j \in [k]}.$$

as we move the root x_i along C . We start computing \tilde{D}_1 by using a shortest-path tree in H and in H_{out} from x_1 . We store the sequence \tilde{D}_1 using the data structure of Lemma 13.

For $i = 2$ to $i = k$, to move from \tilde{D}_{i-1} to \tilde{D}_i we have to perform the operations of Λ'_i (as they are) and the operations of Λ_i with reversed sign. Each operation is done in $O(\log n)$ time per operation using the data structure of Lemma 13. Whenever we obtain the sequence \tilde{D}_i , we query the data structure with MIN to obtain

$$m_i := \min\{\tilde{D}_i\} = \min\{d_{H_{\text{out}}}(x_i, x_j) - d_H(x_i, x_j) \mid j \in [k]\}.$$

If some m_i is negative, then we know that for some $x_j \in V(C)$ we have

$$d_{H_{\text{out}}}(x_i, x_j) < d_H(x_i, x_j),$$

and therefore H is not an isometric subgraph of G . If $m_i \geq 0$ for all $i \in [k]$, then we have

$$\forall i, j \in [k] : d_{H_{\text{out}}}(x_i, x_j) \geq d_H(x_i, x_j).$$

Because $\partial H \subseteq V(C) = \{x_1, \dots, x_k\}$, it follows from Lemma 1 that H is an isometric subgraph in G .

Because of Lemma 14 we spend $O(n \log n)$ time to know which $O(n)$ operations in the sequence of numbers have to be performed. Since each operation in the sequence of numbers takes $O(\log n)$ time (Lemma 13), the total running time is $O(n \log n)$.

A similar procedure can be done for testing *convexity* because of Lemma 4: the graph H is convex in G if and only if $m_i > 0$ for all $i \in [k]$. We summarize.

► **Theorem 15.** *Let G be a plane graph with n vertices and let $H = \text{int}(G, C)$ be a subgraph of G enclosed by a cycle C of G . In $O(n \log n)$ time we can test whether H is an isometric subgraph of G and whether H is a convex subgraph of G .*

The result can be extended to subgraphs of a plane graph that are defined by a few cycles, as shown in Figure 1. We omit this extension from this shortened version.

References

- 1 Stephen Alstrup, Jacob Holm, Kristian de Lichtenberg, and Mikkell Thorup. Maintaining information in fully dynamic trees with top trees. *ACM Trans. Algorithms*, 1(2):243–264, 2005. doi:10.1145/1103963.1103966.

- 2 Hans-Jürgen Bandelt and Victor Chepoi. Graphs with connected medians. *SIAM J. Discret. Math.*, 15(2):268–282, 2002. doi:10.1137/S089548019936360X.
- 3 Hans-Jürgen Bandelt and Victor Chepoi. Metric graph theory and geometry: a survey. In *Surveys on discrete and computational geometry*, volume 453 of *Contemp. Math.*, pages 49–86. Amer. Math. Soc., Providence, RI, 2008. doi:10.1090/conm/453/08795.
- 4 Laurine Bénêteau, Jérémie Chalopin, Victor Chepoi, and Yann Vaxès. Medians in median graphs and their cube complexes in linear time. *J. Comput. Syst. Sci.*, 126:80–105, 2022. doi:10.1016/J.JCSS.2022.01.001.
- 5 Joseph Berleant, Kristin Sheridan, Anne Condon, Virginia Vassilevska Williams, and Mark Bathe. Isometric Hamming embeddings of weighted graphs. *Discret. Appl. Math.*, 332:119–128, 2023. doi:10.1016/J.DAM.2023.02.005.
- 6 Karl Bringmann, Thore Husfeldt, and Måns Magnusson. Multivariate analysis of orthogonal range searching and graph distances. *Algorithmica*, 82(8):2292–2315, 2020. doi:10.1007/s00453-020-00680-z.
- 7 Sergio Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. *ACM Trans. Algorithms*, 15(2):21:1–21:38, 2019. doi:10.1145/3218821.
- 8 Sergio Cabello. Computing the inverse geodesic length in planar graphs and graphs of bounded treewidth. *ACM Trans. Algorithms*, 18(2):14:1–14:26, 2022. doi:10.1145/3501303.
- 9 Sergio Cabello, Erin W. Chambers, and Jeff Erickson. Multiple-source shortest paths in embedded graphs. *SIAM J. Comput.*, 42(4):1542–1571, 2013. doi:10.1137/120864271.
- 10 Sergio Cabello and Christian Knauer. Algorithms for graphs of bounded treewidth via orthogonal range searching. *Comput. Geom.*, 42(9):815–824, 2009. doi:10.1016/j.comgeo.2009.02.001.
- 11 Manoj Changat, Henry Martyn Mulder, and Gerard Sierksma. Convexities related to path properties on graphs. *Discret. Math.*, 290(2/3):117–131, 2005. doi:10.1016/j.disc.2003.07.014.
- 12 Victor D. Chepoi. Isometric subgraphs of Hamming graphs and d-convexity. *Cybernetics (Kibernetika)*, 24:6–11, 1988. doi:10.1007/BF01069520.
- 13 Jongmin Choi, Sergio Cabello, and Hee-Kap Ahn. Maximizing dominance in the plane and its applications. *Algorithmica*, 83(11):3491–3513, 2021. doi:10.1007/s00453-021-00863-2.
- 14 Debarati Das, Evangelos Kipouridis, Maximilian Probst Gutenberg, and Christian Wulff-Nilsen. A simple algorithm for multiple-source shortest paths in planar digraphs. In *5th Symposium on Simplicity in Algorithms, SOSA*, pages 1–11. SIAM, 2022. doi:10.1137/1.9781611977066.1.
- 15 Dragomir Ž. Djoković. Distance-preserving subgraphs of hypercubes. *Journal of Combinatorial Theory, Series B*, 14:263–267, 1973. doi:10.1016/0095-8956(73)90010-5.
- 16 Mitre Costa Dourado, John G. Gimbel, Jan Kratochvíl, Fábio Protti, and Jayme Luiz Szwarcfiter. On the computation of the hull number of a graph. *Discret. Math.*, 309(18):5668–5674, 2009. doi:10.1016/j.disc.2008.04.020.
- 17 Guillaume Ducoffe, Michel Habib, and Laurent Viennot. Diameter, eccentricities and distance oracle computations on H -minor free graphs and graphs of bounded (distance) Vapnik-Chervonenkis dimension. *SIAM J. Comput.*, 51(5):1506–1534, 2022. doi:10.1137/20M136551X.
- 18 David Eppstein. Recognizing partial cubes in quadratic time. *J. Graph Algorithms Appl.*, 15(2):269–293, 2011. doi:10.7155/jgaa.00226.
- 19 V.V. Firsov. Isometric embedding of a graph in a Boolean cube. *Cybernetics (Kibernetika)*, 1:112–113, 1965. doi:10.1007/BF01074705.
- 20 Michael L. Fredman and Robert E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, 1987. doi:10.1145/28869.28874.
- 21 Dmitrii Gavrilov and Ilya Makarov. Fast approximate convex hull construction in networks via node embedding. *IEEE Access*, 11:54588–54595, 2023. doi:10.1109/ACCESS.2023.3281337.
- 22 Paweł Gawrychowski, Haim Kaplan, Shay Mozes, Micha Sharir, and Oren Weimann. Voronoi diagrams on planar graphs, and computing the diameter in deterministic $\tilde{O}(n^{5/3})$ time. *SIAM J. Comput.*, 50(2):509–554, 2021. doi:10.1137/18M1193402.

- 23 Roland Glantz and Henning Meyerhenke. On finding convex cuts in general, bipartite and plane graphs. *Theor. Comput. Sci.*, 695:54–73, 2017. doi:10.1016/j.tcs.2017.07.026.
- 24 Richard Hammack, Wilfried Imrich, and Sandi Klavžar. *Handbook of product graphs*. CRC Press, Boca Raton, FL, second edition, 2011.
- 25 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.
- 26 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.
- 27 Wilfried Imrich and Sandi Klavžar. A convexity lemma and expansion procedures for bipartite graphs. *Eur. J. Comb.*, 19(6):677–685, 1998. doi:10.1006/eujc.1998.0229.
- 28 Wilfried Imrich, Sandi Klavžar, and Henry Martyn Mulder. Median graphs and triangle-free graphs. *SIAM J. Discret. Math.*, 12(1):111–118, 1999. doi:10.1137/S0895480197323494.
- 29 Wilfried Imrich and Iztok Peterin. Recognizing Cartesian products in linear time. *Discret. Math.*, 307(3-5):472–483, 2007. doi:10.1016/j.disc.2005.09.038.
- 30 Adam Karczmarz and Da Wei Zheng. Subquadratic algorithms in minor-free digraphs: (weighted) distance oracles, decremental reachability, and more. In *Proceedings of the 2025 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 4338–4351, 2025. doi:10.1137/1.9781611978322.147.
- 31 Sandi Klavžar and Henry Martyn Mulder. Median graphs: Characterizations, location theory and related structures. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 30:103–127, 1999.
- 32 Philip N. Klein. Multiple-source shortest paths in planar graphs. In *ACM-SIAM Symposium on Discrete Algorithms, SODA 2005*, pages 146–155. SIAM, 2005. URL: <http://dl.acm.org/citation.cfm?id=1070432.1070454>.
- 33 Hung Le and Christian Wulff-Nilsen. VC set systems in minor-free (di)graphs and applications. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 5332–5360, 2024. doi:10.1137/1.9781611977912.192.
- 34 Tilen Marc and Lovro Šubelj. Convexity in complex networks. *Netw. Sci.*, 6(2):176–203, 2018. doi:10.1017/nws.2017.37.
- 35 Henry Martyn Mulder and Alexander Schrijver. Median graphs and Helly hypergraphs. *Discret. Math.*, 25(1):41–50, 1979. doi:10.1016/0012-365X(79)90151-1.
- 36 Martyn Mulder. The structure of median graphs. *Discret. Math.*, 24(2):197–204, 1978. doi:10.1016/0012-365X(78)90199-1.
- 37 Ladislav Nebeský. Median graphs. *Commentationes Mathematicae Universitatis Carolinae*, 12(2):317–325, 1971.
- 38 Sergei Ovchinnikov. *Partial Cubes*, chapter 5, pages 127–181. Springer New York, 2011. doi:10.1007/978-1-4614-0797-3_5.
- 39 Ignacio M. Pelayo. *Geodesic convexity in graphs*. SpringerBriefs in Mathematics. Springer, New York, 2013. doi:10.1007/978-1-4614-8699-2.
- 40 Seth Pettie and Vijaya Ramachandran. Computing shortest paths with comparisons and additions. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 267–276, 2002. URL: <http://dl.acm.org/citation.cfm?id=545381.545417>.
- 41 Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Symposium on Theory of Computing Conference, STOC'13*, pages 515–524. ACM, 2013. doi:10.1145/2488608.2488673.
- 42 Kristin Sheridan, Joseph Berleant, Mark Bathe, Anne Condon, and Virginia Vassilevska Williams. Factorization and pseudofactorization of weighted graphs. *Discret. Appl. Math.*, 337:81–105, 2023. doi:10.1016/J.DAM.2023.04.019.
- 43 Daniel D. Sleator and Robert E. Tarjan. A data structure for dynamic trees. *Journal of Computer and System Sciences*, 26(3):362–391, 1983. doi:10.1016/0022-0000(83)90006-5.

12:16 Testing Whether a Subgraph Is Convex or Isometric

- 44 Lovro Šubelj, Dalibor Fiala, Tadej Ciglarič, and Luka Kronegger. Convexity in scientific collaboration networks. *J. Informetrics*, 13(1):10–31, 2019. doi:10.1016/j.joi.2018.11.005.
- 45 Maximilian Thiessen and Thomas Gaertner. Active learning of convex halfspaces on graphs. In *Advances in Neural Information Processing Systems*, volume 34, pages 23413–23425. Curran Associates, Inc., 2021. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/c4bf1e24f3e6f92ca9dfd9a7a1a1049c-Paper.pdf.
- 46 Mikkel Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *J. ACM*, 46(3):362–394, 1999. doi:10.1145/316542.316548.
- 47 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005. doi:10.1016/j.tcs.2005.09.023.
- 48 Peter M. Winkler. Isometric embedding in products of complete graphs. *Discret. Appl. Math.*, 7(2):221–225, 1984. doi:10.1016/0166-218X(84)90069-6.