Local Certification of Geometric Graph Classes

Oscar Defrain
Aix Marseille Université, CNRS, LIS, Marseille, France

Louis Esperet
Laboratoire G-SCOP (CNRS, Univ. Grenoble Alpes), Grenoble, France

Aurélie Lagoutte
Laboratoire G-SCOP (CNRS, Univ. Grenoble Alpes), Grenoble, France

Pat Morin
School of Computer Science, Carleton University, Canada

Jean-Florent Raymond
CNRS, LIP, ENS de Lyon, France

Abstract

The goal of local certification is to locally convince the vertices of a graph G that G satisfies a given property. A prover assigns short certificates to the vertices of the graph, then the vertices are allowed to check their certificates and the certificates of their neighbors, and based only on this local view and their own unique identifier, they must decide whether G satisfies the given property. If the graph indeed satisfies the property, all vertices must accept the instance, and otherwise at least one vertex must reject the instance (for any possible assignment of certificates). The goal is to minimize the size of the certificates.

In this paper we study the local certification of geometric and topological graph classes. While it is known that in n-vertex graphs, planarity can be certified locally with certificates of size $O(\log n)$, we show that several closely related graph classes require certificates of size $\Omega(n)$. This includes penny graphs, unit-distance graphs, (induced) subgraphs of the square grid, 1-planar graphs, and unit-square graphs. These bounds are tight up to a constant factor and give the first known examples of hereditary (and even monotone) graph classes for which the certificates must have linear size. For unit-disk graphs we obtain a lower bound of $\Omega(n^{1-\delta})$ for any $\delta > 0$ on the size of the certificates, and an upper bound of $O(n \log n)$. The lower bounds are obtained by proving rigidity properties of the considered graphs, which might be of independent interest.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Computing methodologies → Distributed computing methodologies; Theory of computation → Computational geometry

Keywords and phrases Local certification, proof labeling schemes, geometric intersection graphs

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.48

Related Version *Full Version*: `https://arxiv.org/abs/2311.16953` [5]

Funding *Louis Esperet*: partially supported by the French ANR Project TWIN-WIDTH (ANR-21-CE48-0014-01), and by LabEx PERSYVAL-lab (ANR-11-LABX-0025).
Aurélie Lagoutte: partially supported by French ANR project GRALMECO (ANR-21-CE48-0004).
Pat Morin: partially supported by NSERC.
Jean-Florent Raymond: partially supported by French ANR project GRALMECO (ANR-21-CE48-0004).

1 Introduction

Local certification is an emerging subfield of distributed computing where the goal is to assign short certificates to each of the nodes of a network (some connected graph G) such that the nodes can collectively decide whether G satisfies a given property (i.e., whether it belongs

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 48; pp. 48:1–48:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to some given graph class $\mathcal{C}$) by only inspecting their unique identifier, their certificate and the certificates of their neighbors. This assignment of certificates is called a *proof labeling scheme*, and its *complexity* is the maximum number of bits of a certificate (as a function of the number of vertices of G, which is usually denoted by n in the paper). If a graph class $\mathcal{C}$ admits a proof labeling scheme of complexity $f(n)$, we say that $\mathcal{C}$ has *local complexity* $f(n)$. Proof labelling schemes are distributed analogues of traditional non-deterministic algorithms, and graph classes of logarithmic local complexity can be considered as distributed analogues of classes whose recognition is in NP [7]. The notion of proof labeling scheme was formally introduced by Korman, Kutten and Peleg in [17], but originates in earlier work on self-stabilizing algorithms (see again [7] for the history of local certification and a thorough introduction to the field). While every graph class has local complexity $O(n^2)$ [17],[1] the work of [13] identified three natural ranges of local complexity for graph classes:

- $\Theta(1)$: this includes k-colorability for fixed k, and in particular bipartiteness;
- $\Theta(\log n)$: this includes non-bipartiteness and acyclicity; and
- $\Theta(\text{poly}(n))$: this includes non-3-colorability and problems involving symmetry.

It was later proved in [19] that any graph class which can be recognized in linear time (by a centralized algorithm) has an "interactive" proof labeling scheme of complexity $O(\log n)$, where "interactive" means that there are several rounds of interaction between the prover (the entity which assigns certificates) and the nodes of the network (see also [16] for more on distributed interactive protocols). A natural question is whether the interactions are necessary or whether such graph classes have classical proof labeling schemes of complexity $O(\log n)$ as defined above, that is, without multiple rounds of interaction. This question triggered the work of [9] on planar graphs, which have a well-known linear time recognition algorithm. The authors of [9] proved that the class of planar graphs indeed has local complexity $O(\log n)$, and asked whether the same holds for any proper minor-closed class.[2] This was later proved for graphs embeddable on any fixed surface in [10] (see also [6]) and in [2] for classes excluding small minors, while it was proved in [12] that classes excluding a planar graph H as a minor have local complexity $O(\log^2 n)$. The authors of [12] also proved the related result that any graph class of bounded treewidth which is expressible in second order monadic logic has local complexity $O(\log^2 n)$ (this implies in particular that for any fixed k, the class of graphs of treewidth at most k has local complexity $O(\log^2 n)$). Similar meta-theorems involving graph classes expressible in some logic were proved for graphs of bounded treedepth in [8] and graphs of bounded cliquewidth in [11].

Closer to the topic of the present paper, the authors of [14] obtained proof labeling schemes of complexity $O(\log n)$ for a number of classes of geometric intersection graphs, including interval graphs, chordal graphs, circular-arc graphs, trapezoid graphs, and permutation graphs. It was noted earlier in [15] (which proved various results on interactive proof labeling schemes for geometric graph classes) that the "only" classes of graphs for which large lower bounds on the local complexity are known (for instance non-3-colorability, or some properties involving symmetry) are not *hereditary*, meaning that they are not closed under taking induced subgraphs. It turns out that an example of a hereditary class with polynomial local complexity had already been identified in [4] a couple of years earlier: triangle-free graphs (the lower bound on the local complexity given there was sublinear). It was speculated in [15]

[1] Give to every vertex the adjacency matrix of the graph.
[2] Note that it is easy to show that for any minor-closed class $\mathcal{C}$, the complement of $\mathcal{C}$ has local complexity $O(\log n)$, using Robertson and Seymour's Graph Minor Theorem [21].

that any class of geometric intersection graphs has small local complexity, as such classes are both hereditary and well-structured.

Results

In this paper we identify a key rigidity property in graph classes and use it to derive a number of *linear* lower bounds on the local complexity of graph classes defined using geometric or topological properties. These bounds are all best possible, up to $n^{o(1)}$ factors. So our main result is that for a number of classical hereditary graph classes studied in structural graph theory, topological graph theory, and graph drawing, the local complexity is $\Theta(n)$. These are the first non-trivial examples of hereditary classes (some of our examples are even monotone) with linear local complexity. Interestingly, all the classes we consider are very close to the class of planar graphs (which is known to have local complexity $\Theta(\log n)$ [9, 6]): most of these classes are either subclasses or superclasses of planar graphs. Given the earlier results on graphs of bounded treewidth [12] and planar graphs, it is natural to try to understand which sparse graph classes have (poly)logarithmic local complexity. It would have been tempting to conjecture that any (monotone or hereditary) graph class of *bounded expansion* (in the sense of Nešetřil and Ossona de Mendez [20]) has polylogarithmic local complexity, but our results show that this is false, even for very simple monotone classes of linear expansion.

We first show that every class of graphs that contains at most $2^{f(n)}$ unlabeled graphs of size n has local complexity $f(n) + O(\log n)$. This implies all the upper bounds we obtain in this paper, as the classes of graphs we consider usually contain $2^{O(n)}$ or $2^{O(n \log n)}$ unlabeled graphs of size n.

We then turn to lower bounds. Using rigidity properties in the classes we consider, we give a $\Omega(n)$ bound on the local complexity of *penny graphs* (contact graphs of unit-disks in the plane), *unit-distance graphs* (graphs that admit an embedding in $\mathbb{R}^2$ where adjacent vertices are exactly the vertices at Euclidean distance 1), and (induced) subgraphs of the square grid. We then consider *1-planar graphs*, which are graphs admitting a planar drawing in which each edge is crossed by at most one edge. This superclass of planar graphs shares many similarities with them, but we nevertheless prove that it has local complexity $\Theta(n)$ (while planar graphs have local complexity $\Theta(\log n)$).

Next, we consider *unit-square graphs* (intersection graphs of unit-squares in the plane). We obtain a linear lower bound on the local complexity of triangle-free unit-square graphs (which are planar) and of unit-square graphs in general. Finally, we consider *unit-disk graphs* (intersection of unit-disks in the plane), which are widely used in distributed computing as a model of wireless communication networks. For this class we reuse some key ideas introduced in the unit-square case, but as unit-disk graphs are much less rigid we need to introduce a number of new tools, which might be of independent interest in the study of rigidity in geometric graph classes. In particular we answer questions such as: what is asymptotically the minimum number of vertices in a unit-disk graph G such that in any unit-disk embedding of G, two given vertices u and v are at Euclidean distance at least n and at most $n+1$? Or at distance at least n and at most $n+\varepsilon$, for $\varepsilon \ll n$? Using our constructions we obtain a lower bound of $\Omega(n^{1-\delta})$ (for every $\delta > 0$) on the local complexity of unit-disk graphs. As there are at most $2^{O(n \log n)}$ unlabelled unit-disk graphs on n vertices [18], our first result implies that the local complexity of unit-disk graphs is $O(n \log n)$, so our results are nearly tight.

Techniques

All our lower bounds are inspired by the set-disjointness problem in non-deterministic communication complexity. This approach was already used in earlier work in local certification, in order to provide lower bounds on the local complexity of computing the diameter [3], or for certifying non-3-colorability [13]. Here the main challenge is to translate the technique into geometric constraints. The key point of the set-disjointness problem is informally the following: let $A, B \subseteq \{1, \ldots, N\}$ be the input of some kind of "two-party system" that must decide whether A and B are disjoint, given that one party knows A and the other knows B; then at least N bits of shared (or exchanged) information are necessary for them to decide correctly. Otherwise, there are fewer bit combinations than the 2^N entries of the form $(A, \overline{A})$, hence the two parties can be fooled to accept a negative instance built from two particular positive instances sharing the same bit combination. In the setting of non-deterministic communication complexity, the two parties are Alice and Bob; in our setting, the two parties will be two subsets of vertices covering the graph and with small intersection (the intersection must be a small cutset of the whole graph): in the following, we refer to those two connected subsets of vertices as respectively the "left" part and the "right" part of the graph. The "shared" bits of information will be the certificates given to their intersection (and to its neighborhood). To express the sets A, B and their disjointness, the left (resp. right) part of the graph will be equipped with a path P_A (resp. P_B) of length $\Omega(N)$, such that P_A and P_B only intersect in their endpoints.[3] The crucial rigidity property which we will require is that in any embedding of G as a geometric graph from some class $\mathcal{C}$, the two paths P_A and P_B will be very close, in the sense that if $P_A = a_1, \ldots, a_\ell$ and $P_B = b_1, \ldots, b_\ell$, then a_i is close to b_i for any $1 \leq i \leq \ell$. Using this property, we will attach some gadgets to the vertices of the path P_A (resp. P_B) depending on A (resp. B), in such a way that the resulting graph lies in the class $\mathcal{C}$ if and only if A and B are disjoint. As there is little connectivity between the left and the right part, the endpoints of the paths will have to contain very long certificates in order to decide whether A and B are disjoint, hence whether $G \in \mathcal{C}$ or not.

We present the results in increasing order of difficulty. Subgraphs or induced subgraphs of infinite graphs such as grids are perfectly rigid in some sense, with some graphs having unique embeddings up to symmetry. Unit-square graphs are much less rigid but we can use nice properties of the ℓ_∞-distance and the uniqueness of embeddings of 3-connected planar graphs. We conclude with unit-disk graphs, which is the least rigid class we consider. The Euclidean distance misses most of the properties enjoyed by the ℓ_∞-distance and we must work much harder to obtain the desired rigidity property.

Outline

We start with some preliminaries on graph classes and local certification in Section 2. We prove our general upper bound result in Section 3. Section 4 introduces the notion of *disjointness-expressing* class of graphs, highlighting the key properties needed to derive our local certification lower bounds. We deduce in Section 5 linear lower bounds on the local complexity of subgraphs of the grid, penny graphs, and 1-planar graphs. Section 6 is devoted for the linear lower bound on the local complexity of unit-square graphs, while Section 7 contains our main result, a quasi-linear lower bound on the local complexity of unit-disk graphs. We conclude in Section 8 with a number of questions and open problems.

Due to the limit on the number of pages of the submission, most of the proofs have been omitted in this version. They are available in the full version of the paper [5].

---

[3] We note here that the proof for 1-planar graphs diverges from this approach, but it is the only one.

2 Preliminaries

In this paper logarithms are binary, and graphs are assumed to be simple, loopless, undirected, and connected. The *length* of a path P, denoted by $|P|$, is the number of edges of P. The *distance* between two vertices u and v in a graph G, denoted by $d_G(u, v)$ is the minimum length of a path between u and v. The *neighborhood* of a vertex v in a graph G, denoted by $N_G(v)$ (or $N(v)$ if G is clear from the context), is the set of vertices at distance exactly 1 from v. The *closed neighborhood* of v, denoted by $N_G[v] := \{v\} \cup N_G(v)$, is the set of vertices at distance at most 1 from v. For a set S of vertices of G, we define $N_G[S] := \bigcup_{v \in S} N_G[v]$.

2.1 Local certification

The vertices of any n-vertex graph G are assumed to be assigned distinct (but otherwise arbitrary) identifiers $(\mathrm{id}(v))_{v \in V(G)}$ from $\{1, \dots, \mathrm{poly}(n)\}$. When we refer to a subgraph H of a graph G, we implicitly refer to the corresponding labeled subgraph of G. Note that the identifiers of each of the vertices of G can be stored using $O(\log n)$ bits, where log denotes the binary logarithm. We follow the terminology introduced by Göös and Suomela [13].

Proofs and provers

A *proof* for a graph G is a function $P : V(G) \to \{0, 1\}^*$ (as G is a labeled graph, the proof P is allowed to depend on the identifiers of the vertices of G). The binary words $P(v)$ are called *certificates*. The *size* of P is the maximum size of a certificate $P(v)$, for $v \in V(G)$. A *prover* for a graph class $\mathcal{G}$ is a function that maps every $G \in \mathcal{G}$ to a proof for G.

Local verifiers

A *verifier* $\mathcal{A}$ is a function that takes a graph G, a proof P for G, and a vertex $v \in V(G)$ as inputs, and outputs an element of $\{0, 1\}$. We say that v *accepts* the instance if $\mathcal{A}(G, P, v) = 1$ and that v *rejects* the instance if $\mathcal{A}(G, P, v) = 0$.

Consider a graph G, a proof P for G, and a vertex $v \in V(G)$. We denote by $G[v]$ the subgraph of G induced by $N[v]$, the closed neighborhood of v, and similarly we denote by $P[v]$ the restriction of P to $N[v]$.

A verifier $\mathcal{A}$ is *local* if for any $v \in G$, the output of v only depends on its identifier and $P[v]$.

Note that our lower bounds hold in the stronger model of *locally checkable proofs* of Göös and Suomela [13], where in addition the output of v is allowed to depend on $G[v]$, that is $\mathcal{A}(G, P, v) = \mathcal{A}(G[v], P[v], v)$ for any vertex v of G.

Proof labeling schemes

A *proof labeling scheme* for a graph class $\mathcal{G}$ is a prover-verifier pair $(\mathcal{P}, \mathcal{A})$ where $\mathcal{A}$ is local, with the following properties.

Completeness: If $G \in \mathcal{G}$, then $P := \mathcal{P}(G)$ is a proof for G such that for any vertex $v \in V(G)$, $\mathcal{A}(G, P, v) = 1$.

Soundness: If $G \notin \mathcal{G}$, then for every proof P' for G, there exists a vertex $v \in V(G)$ such that $\mathcal{A}(G, P', v) = 0$.

In other words, upon looking at its closed neighborhood (labeled by the identifiers and certificates), the local verifier of each vertex of a graph $G \in \mathcal{G}$ accepts the instance, while if $G \notin \mathcal{G}$, for every possible choice of certificates, the local verifier of at least one vertex rejects the instance.

The *complexity* of the proof labeling scheme is the maximum size of a proof $P = \mathcal{P}(G)$ for an n-vertex graph $G \in \mathcal{G}$, and the *local complexity* of $\mathcal{G}$ is the minimum complexity of a proof labeling scheme for $\mathcal{G}$. If we say that the complexity is $O(f(n))$, for some function f, the $O(\cdot)$ notation refers to $n \to \infty$. See [7, 13] for more details on proof labeling schemes and local certification in general.

2.2 Geometric graph classes

In this section we collect some useful properties that are shared by most of the graph classes we will investigate in the paper.

A *unit-disk graph* (respectively *unit-square graph*) is the intersection graph of unit-disks (respectively unit-squares) in the plane. That is, G is a unit-disk graph if every vertex of G can be mapped to a unit-disk in the plane so that two vertices are adjacent if and only if the corresponding disks intersect, and similarly for squares. A *penny graph* is the contact graph of unit-disks in the plane, i.e., in the definition of unit-disk graphs above we additionally require the disks to be pairwise interior-disjoint. A *unit-distance* graph is a graph whose vertices are points in the plane, where two points are adjacent if and only if their Euclidean distance is equal to 1. Unit-distance graphs clearly form a superclass of penny graphs.

A *drawing* of a graph G in the plane is a mapping from the vertices of G to distinct points in the plane and from the edges of G to Jordan curves, such that for each edge uv in G, the curve associated to uv joins the images of u and v and does not contain the image of any other vertex of G. A graph is *planar* if it has a drawing in the plane with no edge crossings (such a drawing will also be called a *planar graph drawing* in the remainder). Every planar graph drawing of a graph G gives a clockwise cyclic ordering of the neighbors around each vertex of G. We say that that two planar graph drawings of G are *equivalent* if the corresponding cyclic orderings are the same. A *planar graph embedding* of a graph G is an equivalence class of planar graph drawings of G. Given a planar graph embedding of a graph G, all the corresponding (equivalent) planar drawings of G have the same set of faces (but different choices of outerface yield different planar drawings).

A graph is *1-planar* if it has a drawing in the plane such that for each edge e of G, there is at most one edge e' of G distinct from e such that the interior of the curve associated to e intersects the interior of the curve associated to e'.

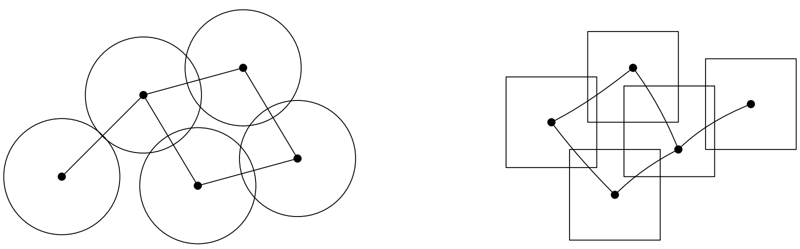

Figure 1 Triangle-free intersection graphs of unit-disks and unit-squares in the plane, and the associated planar graph embeddings.

The following well-known proposition will be useful (see Figure 1 for an illustration).

▶ **Proposition 2.1.** *Consider a family of unit-disks or a family of unit-squares in the plane, and assume that the intersection graph G of the family is connected and triangle-free. Then G is planar, and moreover each representation of G as such an intersection graph of unit-disks or unit-squares in the plane gives rise to a planar graph embedding of G in a natural way (see for instance Figure 1). Furthermore, the representation of G as an intersection graph (of unit-disks or unit-squares) and the resulting planar graph embedding are equivalent, in the sense that the clockwise cyclic ordering of the neighbors around each vertex is the same.*

We will often need to argue that some planar graphs have unique planar embeddings. The following classical result of Whitney will be crucial.

▶ **Theorem 2.2** ([22]). *If a planar graph G is 3-connected (or can be obtained from a 3-connected simple graph by subdividing some edges), then it has a unique planar graph embedding, up to the reversal of all cyclic orderings of neighbors around the vertices.*

We note that the reversal of all cyclic orderings in the statement of Theorem 2.2 corresponds to a reflection of the corresponding planar drawings.

3 Linear upper bounds for tiny classes

Given a class of graphs $\mathcal{C}$ and a positive integer n, let $\mathcal{C}_n$ be the set of all unlabeled graphs of $\mathcal{C}$ having exactly n vertices (i.e., we consider graphs up to isomorphism).

If there is a constant $c > 0$ such that for every positive integer n, $|\mathcal{C}_n| \leq c^n$, then the class $\mathcal{C}$ is said to be *tiny*. This is the case for all proper minor-closed classes (for instance planar graphs). On the other hand, unit-interval graphs and unit-disk graphs do not form tiny classes as proved in [18]. The local complexity and the number of unlabeled graphs in a class are related by the following simple result.

▶ **Theorem 3.1** ([5]). *Any class $\mathcal{C}$ of connected graphs has local complexity at most $\log(|\mathcal{C}_n|) + O(\log n)$. In particular if $\mathcal{C}$ is a tiny class, then the local complexity is $O(n)$.*

Proof (sketch). The prover gives each vertex v the same description of G (as an unlabelled graph, so using $\log(|\mathcal{C}_n|)$ bits), together with the name of the image of v in this description, and the number n of vertices of G. Each vertex checks that its neighborhood is consistent with its image in the description of G, and if so the graph under consideration must have a locally bijective homomorphism to G. The vertices then check that the number of vertices of the graph is indeed $n = |V(G)|$, which implies that this locally bijective homomorphism is an isomorphism to G, as desired. ◀

As a consequence, we immediately obtain the following.

▶ **Corollary 3.2.** *The following classes have local complexity $O(n)$: the class of all (induced) subgraphs of the square grid, penny graphs, 1-planar graphs, triangle-free unit-square graphs, and triangle-free unit-disk graphs.*

The next result directly follows from a bound of order $2^{O(n \log n)}$ on the number of unit-square graphs and unit-disk graphs [18], and on the number of unit-distance graphs [1].

▶ **Corollary 3.3.** *The classes of unit-distance graphs, unit-square graphs, and unit-disk graphs have local complexity $O(n \log n)$.*

The remainder of the paper consists in proving lower bounds of order $\Omega(n)$ (or $\Omega(n^{1-\delta})$, for any $\delta > 0$), for all the classes mentioned in Corollaries 3.2 and 3.3, except triangle-free unit-disk graphs (our quasi-linear lower bound only applies to unit-disk graphs).

4 Disjointness-expressing graph classes

In this section we describe the framework relating the disjointness problem to proof labeling schemes. Our main source of inspiration is [13], where a lower bound on the local complexity of non-3-colorability is proved using a similar approach, and [3] where an explicit reduction to the non-deterministic communication complexity of the disjointness problem is used.

Here we adapt the disjointness problem to fit in our local certification setting. A class $\mathcal{C}$ of graphs is said to be (s, κ)-*disjointness-expressing* if for some constant $\alpha > 0$, for every positive integer N and every $X \subseteq \{1, \ldots, N\}$, one can define graphs $L(X)$ (referred to as the "left part") and $R(X)$ ("right part"), each containing a labeled set S of special vertices such that for every $A, B \subseteq \{1, \ldots, N\}$ the following holds:

1. the graph $g(L(A), R(B))$ obtained by identifying vertices of S in $L(A)$ to the corresponding vertices of S in $R(B)$ is connected and has at most $\alpha N^{1/\kappa}$ vertices;
2. the subgraph of $g(L(A), R(B))$ induced by the closed neighborhood $N_{g(L(A),R(B))}[S]$ of S is independent[4] of the choice of A and B and has at most s vertices; and
3. $g(L(A), R(B))$ belongs to $\mathcal{C}$ if and only if $A \cap B = \emptyset$.

The idea is that S is a small cutset between vertices of $L(A)$, having information on A, and vertices of $R(B)$, having information on B. Deciding whether the graph $g(L(A), R(B))$ belongs to $\mathcal{C}$ amounts to deciding whether A and B are disjoint, which requires N bits of information even in a non-deterministic setting, thus the small cutset at the frontier between $L(A)$ and $R(B)$ must receive long certificates. Otherwise, there are fewer bit combinations at the frontier than the 2^N entries of the form $(A, \overline{A})$, hence the vertices can be fooled to accept a negative instance built from two particular positive instances sharing the same bit combination.

The role of s and κ is explained by the result below.

▶ **Theorem 4.1** ([5]). *Let $\mathcal{C}$ be a (s, κ)-disjointness-expressing class of graphs. Then any proof labeling scheme for the class $\mathcal{C}$ has complexity $\Omega\left(\frac{n^\kappa}{s}\right)$. In particular if s is a constant and $\kappa = 1$, the complexity is $\Omega(n)$.*

Proof (sketch). Let $(\mathcal{P}, \mathcal{A})$ be a proof labeling scheme for the class $\mathcal{C}$ and let $p\colon \mathbb{N} \to \mathbb{N}$ be its complexity. For every $A \subseteq \{1, \ldots, N\}$, let $G_A = g(L(A), R(\overline{A}))$. Clearly $G_A \in \mathcal{C}$ so the verifier $\mathcal{A}$ accepts the proof $P_A = P(G_A)$ on every vertex of G_A. Let n denote the maximum order of G_A for $A \subseteq \{1, \ldots, N\}$.

There are 2^N choices for the set A. On the other hand, in G_A there are at most $2^{s \cdot p(n)}$ different ways to assign certificates to the vertices of $N[S]$. By the Pigeonhole Principle, if $2^N > 2^{sp(n)}$ there are two sets $A, A' \subseteq \{1, \ldots, N\}$ such that the proofs P_A and $P_{A'}$ coincide on the subgraph of G_A and $G_{A'}$ induced by $N[S]$. Since $A \neq A'$, we may assume without loss of generality that $A \cap \overline{A'} \neq \emptyset$. So the graph $G = g(L(A), R(\overline{A'}))$ does not belong to $\mathcal{C}$. We now consider a proof P for G defined as follows: if $v \in V(L(A))$ then $P(v) := P_A(v)$ and if $v \in V(R(\overline{A'}))$ then $P(v) := P_{A'}(v)$. The verifier $\mathcal{A}$ will accept P on every vertex of G, contradiction. Therefore $2^N \leq 2^{s \cdot p(n)}$. Recall that $n \leq \alpha N^{1/\kappa}$, by the definition of disjointness-expressibility. Hence $p(n) = \Omega(n^\kappa/s)$, as claimed. ◀

---

[4] i.e., for every $A, A', B, B' \subseteq \{1, \ldots, N\}$ there is an isomorphism from $g(L(A), R(B))[N_{g(L(A),R(B))}[S]]$ to $g(L(A'), R(B'))[N_{g(L(A'),R(B'))}[S]]$ that is the identity on S.

5 Linear lower bounds in rigid classes

In this section we obtain linear lower bounds on the local complexity of several graph classes using the framework described in Section 4. We only sketch the argument in the case of penny graphs.

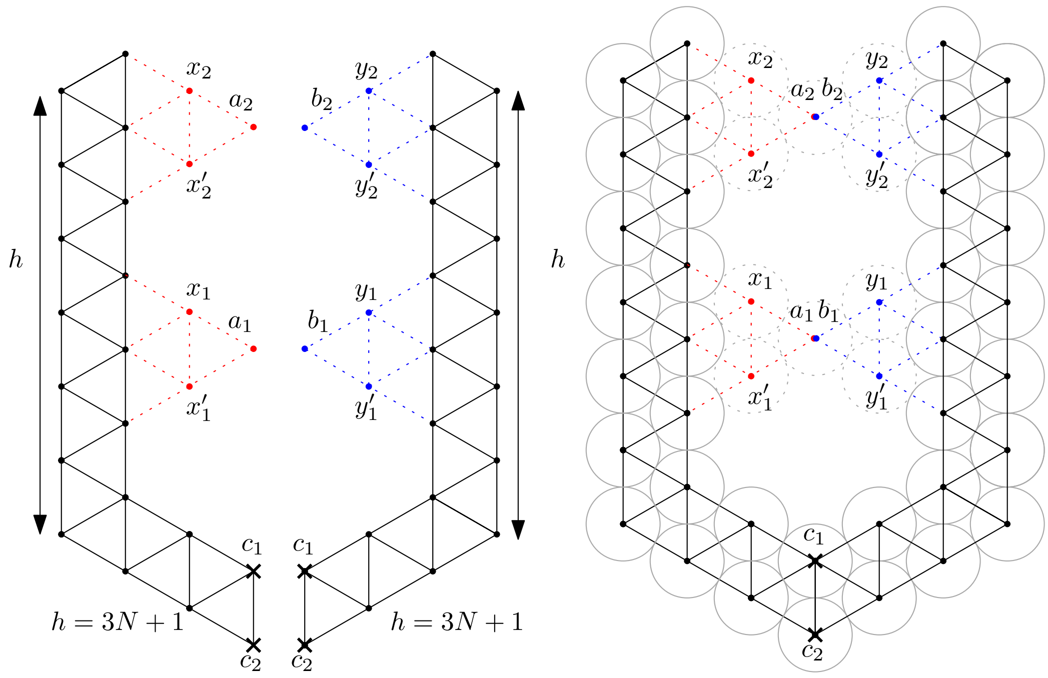

Figure 2 Construction of L, R and g for penny graphs in the case where $N = 2$, with $A, B \subseteq \{1, \dots, N\}$. Color red highlights vertices and edges that depend on the choice of A, and color blue highlights vertices and edges that depend on the choice of B.

▶ **Theorem 5.1** ([5]). *The class of penny graphs is* $(6, 1)$*-disjointness-expressing.*

Proof (sketch). The proof is illustrated in Figure 2. The graph $g(L(A), R(B))$ (on the right) is obtained by identifying each c_i $(i = 1, 2)$ in $L(A)$ with the corresponding vertex in $R(B)$. The graphs $L(A)$ and $R(B)$ are depicted on the left. Vertices a_j, x_j, x'_j are added to the graph $L(A)$ if and only if $j \in A$, and similarly vertices b_j, y_j, y'_j are added to the graph $R(B)$ if and only if $j \in B$. The crucial properties of the construction are that (1) the graph without the added gadgets has a unique embedding as a penny graph, and (2) there is a small cut separating the left and right parts ($\{c_1, c_2\}$, which is far from the gadgets on both sides), and (3) if two gadgets a_i, x_i, x'_i and b_j, y_j, y'_j are added with $i = j$, then the graph is not a penny graph. The last item follows from the fact that a_i and b_j would be mapped to the same point in the plane, and thus a_i would also need to be adjacent to y_j and y'_j in the graph (which they are not). ◀

From Theorems 5.1 and 4.1, together with Corollary 3.2, we immediately deduce the following.

▶ **Theorem 5.2.** *The local complexity of the class of penny graphs is* $\Theta(n)$.

Using our framework we obtain the following results (whose proofs are available in appendix).

▶ **Theorem 5.3.** *The class of unit-distance graphs is* $(6,1)$*-disjointness-expressing.*

▶ **Theorem 5.4** ([5])**.** *The class of subgraphs of the square grid is* $(6,1)$*-disjointness-expressing.*

▶ **Theorem 5.5** ([5])**.** *The class of 1-planar graphs is* $(20,1)$*-disjointness-expressing.*

We immediately deduce the following.

▶ **Theorem 5.6.** *The classes of subgraphs of the square grid, unit-distance graphs and 1-planar graphs all have local complexity* $\Theta(n)$*.*

6 Unit-square graphs

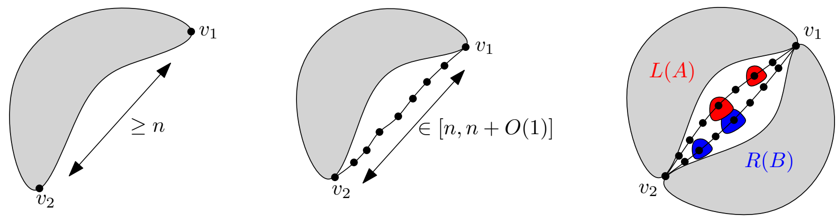

Figure 3 A sketch of the general approach to prove Theorems 6.1 and 7.1.

The graph we constructed in the previous section had some perfect rigidity properties: if the images of a constant number of vertices in the plane were fixed, then the whole graph had at most one embedding in the plane. This does not hold in unit-square graphs, but for our framework it is enough to make sure that once a constant number of vertices are fixed, each vertex of the graph can only be mapped to a small fixed region in any embedding. More precisely, we construct for any n a unit-square graph with $O(n)$ vertices with two specific vertices v_1, v_2 that are at ℓ_∞-distance at least n in any embedding. We add a shortest path connecting v_1 and v_2, so that the resulting graph is still a unit-square graph with $O(n)$ vertices, and the ℓ_∞-distance between v_1 and v_2 is at least n and at most $n + O(1)$ in any embedding. This is illustrated in Figure 3 above, where the path between v_1 and v_2 is close from being mapped to the line segment between the image of v_1 and the image of v_2. This path is then used as an interface to add gadgets expressing any set A for part $L(A)$ and any set B for part $R(B)$, very much as in the proof of Theorem 5.1. The difficulty lies in proving this approximate rigidity property, which follows from the rigidity of the ℓ_∞ norm.

▶ **Theorem 6.1** ([5])**.** *The classes of triangle-free unit-square graphs and unit-square graphs are both* $(6,1)$*-disjointness-expressing.*

Using Theorem 4.1, together with Corollaries 3.2 and 3.3, we immediately deduce the following.

▶ **Theorem 6.2.** *The local complexity of the class of triangle-free unit-square graphs is* $\Theta(n)$*, and the local complexity of the class of unit-square graphs is* $\Omega(n)$ *and* $O(n \log n)$*.*

We note that the proof approach of Theorem 6.1 naturally extends to higher dimension.

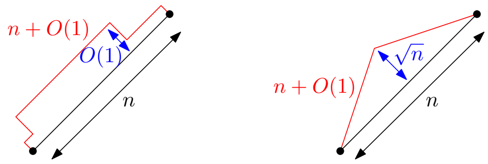

Figure 4 The difference between the ℓ_∞-distance (left) and the ℓ_2-distance (right).

7 Unit-disk graphs

We would like to prove a variant of Theorem 6.1 for unit-disk graphs, but there are two major obstacles. The first is that there does not seem to be a simple unit-disk graph with $O(n)$ vertices with two specified vertices that are at Euclidean distance at least n in any unit-disk embedding. Our construction of such a graph will be significantly more involved (and thus the number of vertices will be only upper bounded by $O(n^{1+\varepsilon})$ for any $\varepsilon > 0$, rather than $O(n)$). The second obstacle comes from Pythagoras' theorem: In the unit-square case, if we consider a path P of length $n + O(1)$ between two vertices u, v embedded in the plane such that their x- and y-coordinates both differ by exactly n, then in any unit-square embedding of P, the vertices of P deviate by at most a constant from the line segment $[u, v]$ between u and v. This is what we used in the proof of Theorem 6.1 to make sure that $L(A)$ and $R(B)$ are so close that the i-th gadget cannot exist both on $L(A)$ and $R(B)$ simultaneously when $i \in A \cap B$. However, as illustrated in Figure 4, Pythagoras' theorem implies that in the Euclidean case, when the Euclidean distance between u and v is equal to n, the vertices of P can deviate by $\Theta(\sqrt{n})$ from the line segment $[u, v]$, which is too much for our purpose (we need a constant deviation). So we need different ideas to make sure the gadgets are embedded sufficiently close to each other.

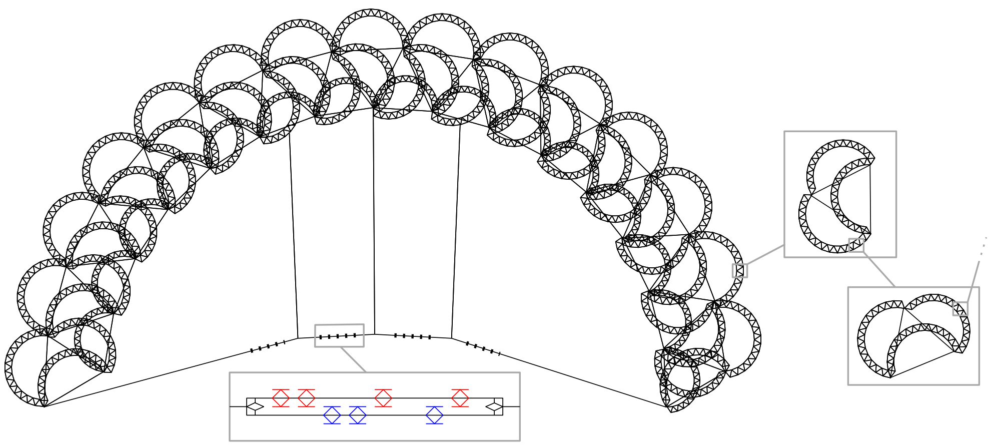

Figure 5 A summary of the construction used in the proof of Theorem 7.1.

The main result of this paper is the following.

▶ **Theorem 7.1** ([5]). *For any $\delta > 0$, the class of unit-disk graphs is $(O(\log n), 1-\delta)$-disjointness-expressing.*

Proof (sketch). A summary of our construction is depicted in Figure 5. The first component of the construction is the arch-shaped part, which is defined recursively and has the property that its endpoints lie at Euclidean distance $\Omega(n)$ in any unit-disk embedding, while the graph only contains $O(n^{1+\varepsilon})$ vertices (for any $\varepsilon > 0$). We then add a shortest possible path P between the endpoints of the arch (with the condition that the resulting graph is still a unit-disk graph) and we would like to argue that the path P is *tight*, in the sense that in any embedding, all the unit-disks corresponding to the vertices of P lie at distance $O(1)$ from the line segment between the endpoints of P. For this we need to add a number of paths connecting P to the arch to make P even tighter. These paths delimit subpaths of P and the construction forces *at least one of* those subpaths to be tight. Since we do not know which subpath will be tight, we add gadgets, called *decorated corridors*, along all these subpaths. When a subpath is tight, the corresponding corridor is sufficiently narrow so that gadgets of $L(A)$ and $R(B)$ along the corridor can emulate the disjointness problem between A and B, as in the proof of Theorem 5.1. There is a gadget of $L(A)$ at position j of every corridor if and only if $j \in A$ and similarly for $R(B)$, and the gadgets of $L(A)$ and $R(B)$ intersect at position j of some corridor if and only if $j \in A \cap B$. Since these gadgets are not adjacent in the graph, this shows that the graph is a unit-disk graph if and only if $A \cap B = \emptyset$, as desired. ◀

Using Theorem 4.1, together with Corollary 3.3, we immediately deduce the following.

▶ **Theorem 7.2.** *The local complexity of the class of unit-disk graphs is $O(n \log n)$ and $\Omega(n^{1-\delta})$ for any $\delta > 0$.*

8 Open problems

In this paper we have obtained a number of optimal (or close to optimal) results on the local complexity of geometric graph classes. Our proofs are based on a new notion of rigidity. It is natural to ask which other graph classes enjoy similar properties. A natural candidate is the class of segment graphs (intersection graphs of line segments in the plane), which have several properties in common with unit-disk graphs. In particular the recognition problems for these classes are complete for the existential theory of the reals, and the minimum bit size for representing an embedding of some of these graphs in the plane is at least exponential in their number of vertices. We believe that the local complexity of segment graphs (and that of the more general class of string graphs) is at least polynomial in their number of vertices. More generally, is it true that all classes of graphs for which the recognition problem is hard for the existential theory of the reals have polynomial local complexity?

It might also be interesting to investigate the smaller class of *circle graphs* (intersection graphs of chords of a circle). The authors of [14] proved that the closely related class of permutation graphs has logarithmic local complexity. It is quite possible that the same holds for circle graphs. See [15] for results on interactive proof labeling schemes for this class and related classes.

We proved that 1-planar graphs have local complexity $\Theta(n)$. What can we say about the local complexity of other graph classes defined with constrained on their drawings in the plane? For instance is it true that for every $k \geq 2$, the local complexity of the class of graphs with queue number at most k is polynomial? What about graphs with stack number at most k?

We have given the first example of non-trivial hereditary (and even monotone) classes of local complexity $\Omega(n)$. Can this be improved? Are there hereditary (or even monotone) classes of local complexity $\Omega(n^c)$ for $c > 1$?

References

1 Noga Alon and Andrey Kupavskii. Two notions of unit distance graphs. *Journal of Combinatorial Theory, Series A*, 125:1–17, 2014.

2 Nicolas Bousquet, Laurent Feuilloley, and Théo Pierron. Local certification of graph decompositions and applications to minor-free classes. In Quentin Bramas, Vincent Gramoli, and Alessia Milani, editors, *25th International Conference on Principles of Distributed Systems, OPODIS 2021, December 13-15, 2021, Strasbourg, France*, volume 217 of *LIPIcs*, pages 22:1–22:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.OPODIS.2021.22.

3 Keren Censor-Hillel, Ami Paz, and Mor Perry. Approximate proof-labeling schemes. *Theoretical Computer Science*, 811:112–124, 2020. doi:10.1016/j.tcs.2018.08.020.

4 Pierluigi Crescenzi, Pierre Fraigniaud, and Ami Paz. Trade-offs in distributed interactive proofs. In Jukka Suomela, editor, *33rd International Symposium on Distributed Computing, DISC 2019, October 14-18, 2019, Budapest, Hungary*, volume 146 of *LIPIcs*, pages 13:1–13:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICS.DISC.2019.13.

5 Oscar Defrain, Louis Esperet, Aurélie Lagoutte, Pat Morin, and Jean-Florent Raymond. Local certification of geometric graph classes. *CoRR*, abs/2311.16953, 2023. arXiv:2311.16953, doi:10.48550/arXiv.2311.16953.

6 Louis Esperet and Benjamin Lévêque. Local certification of graphs on surfaces. *Theoretical Computer Science*, 909:68–75, 2022. doi:10.1016/j.tcs.2022.01.023.

7 Laurent Feuilloley. Introduction to local certification. *Discrete Mathematics & Theoretical Computer Science*, 23(3), 2021. arXiv:1910.12747.

8 Laurent Feuilloley, Nicolas Bousquet, and Théo Pierron. What can be certified compactly? Compact local certification of MSO properties in tree-like graphs. In Alessia Milani and Philipp Woelfel, editors, *PODC '22: ACM Symposium on Principles of Distributed Computing, Salerno, Italy, July 25 - 29, 2022*, pages 131–140. ACM, 2022. doi:10.1145/3519270.3538416.

9 Laurent Feuilloley, Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, Éric Rémila, and Ioan Todinca. Compact distributed certification of planar graphs. *Algorithmica*, 83(7):2215–2244, 2021. doi:10.1007/s00453-021-00823-w.

10 Laurent Feuilloley, Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, Éric Rémila, and Ioan Todinca. Local certification of graphs with bounded genus. *Discrete Applied Mathematics*, 325:9–36, 2023. doi:10.1016/j.dam.2022.10.004.

11 Pierre Fraigniaud, Frédéric Mazoit, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. Distributed certification for classes of dense graphs. *arXiv preprint*, 2023. arXiv:2307.14292.

12 Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. A meta-theorem for distributed certification. In Merav Parter, editor, *Structural Information and Communication Complexity - 29th International Colloquium, SIROCCO 2022, Paderborn, Germany, June 27-29, 2022, Proceedings*, volume 13298 of *Lecture Notes in Computer Science*, pages 116–134. Springer, 2022. doi:10.1007/978-3-031-09993-9_7.

13 Mika Göös and Jukka Suomela. Locally checkable proofs in distributed computing. *Theory of Computing*, 12(1):1–33, 2016. doi:10.4086/toc.2016.v012a019.

14 Benjamin Jauregui, Pedro Montealegre, Diego Ramírez-Romero, and Ivan Rapaport. Local certification of some geometric intersection graph classes. *CoRR*, abs/2309.04789, 2023. doi:10.48550/arXiv.2309.04789.

15 Benjamin Jauregui, Pedro Montealegre, and Ivan Rapaport. Distributed interactive proofs for the recognition of some geometric intersection graph classes. In Merav Parter, editor, *Structural Information and Communication Complexity - 29th International Colloquium, SIROCCO 2022, Paderborn, Germany, June 27-29, 2022, Proceedings*, volume 13298 of *Lecture Notes in Computer Science*, pages 212–233. Springer, 2022. doi:10.1007/978-3-031-09993-9_12.

16 Gillat Kol, Rotem Oshman, and Raghuvansh R. Saxena. Interactive distributed proofs. In Calvin Newport and Idit Keidar, editors, *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, pages 255–264. ACM, 2018. URL: https://dl.acm.org/citation.cfm?id=3212771.

17 Amos Korman, Shay Kutten, and David Peleg. Proof labeling schemes. *Distributed Computing*, 22(4):215–233, 2010. doi:10.1007/s00446-010-0095-3.

18 Colin McDiarmid and Tobias Müller. The number of disk graphs. *European Journal of Combinatorics*, 35:413–431, 2014. Selected Papers of EuroComb'11. doi:10.1016/j.ejc.2013.06.037.

19 Moni Naor, Merav Parter, and Eylon Yogev. The power of distributed verifiers in interactive proofs. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1096–115. SIAM, 2020. doi:10.1137/1.9781611975994.67.

20 Jaroslav Nešetřil and Patrice Ossona De Mendez. *Sparsity: graphs, structures, and algorithms*, volume 28. Springer Science & Business Media, 2012.

21 Neil Robertson and Paul D. Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004. doi:10.1016/j.jctb.2004.08.001.

22 Hassler Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55(1):245–254, 1933.