Subexponential algorithms for variants of homomorphism problem in string graphs

Karolina Okrasa and Paweł Rzążewski

Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, Poland
okrasak@student.mini.pw.edu.pl, p.rzazewski@mini.pw.edu.pl

Abstract

We consider the complexity of finding weighted homomorphisms from intersection graphs of curves (string graphs) with n vertices to a fixed graph H. We provide a complete dichotomy for the problem: if H has no two vertices sharing two common neighbors, then the problem can be solved in time \(2^{O(n^{2/3} \log n)}\), otherwise there is no algorithm working in time \(2^{o(n)}\), even in intersection graphs of segments, unless the ETH fails. This generalizes several known results concerning the complexity of computational problems in geometric intersection graphs.

Then we consider two variants of graph homomorphism problem, called locally injective homomorphism and locally bijective homomorphism, where we require the homomorphism to be injective or bijective on the neighborhood of each vertex. We show that for each target graph H, both problems can always be solved in time \(2^{O(\sqrt{n \log n})}\) in string graphs.

For the locally surjective homomorphism, defined in an analogous way, the situation seems more complicated. We show the dichotomy theorem for simple connected graphs H with maximum degree 2. If H is isomorphic to \(P_3\) or \(C_4\), then the existence of a locally surjective homomorphism from a string graph with n vertices to H can be decided in time \(2^{O(n^{2/3} \log^{3/2} n)}\), otherwise the problem cannot be solved in time \(2^{o(n)}\), unless the ETH fails.

As a byproduct, we obtain several results concerning the complexity of variants of homomorphism problem in \(P_t\)-free graphs. In particular, we obtain the dichotomy theorem for weighted homomorphism, analogous to the one for string graphs.

1 Introduction

Theory of NP-completeness gives us tools to identify problems, which are unlikely to admit polynomial-time algorithms, but it does not give any insight into possible complexities of problems that are considered hard. For example, the best algorithms we know for most canonical problems like 3-COLORING, INDEPENDENT SET, DOMINATING SET, VERTEX COVER, HAMILTONIAN CYCLE, are single-exponential, i.e., with complexity \(2^{O(n)}\) (n will always denote the number of vertices in the input graph). On the other hand, in planar graphs these problems are still NP-complete, but they admit subexponential algorithms (i.e., working in time \(2^{o(n)}\)). Indeed, most canonical problems in planar graphs admit a certain “square-root phenomenon”, i.e., can be solved in time \(2^{O(\sqrt{n})}\). The core building block in construction of subexponential algorithms for planar graphs is the celebrated planar decomposition theorem by Lipton and Tarjan [35], which asserts that every planar graph has a balanced separator of size \(O(\sqrt{n})\).

To argue whether those algorithms are asymptotically optimal and, in general, to prove meaningful lower bounds on the complexity of hard problems, we need a stronger assumption than “P ≠ NP”. Such a stronger assumption, commonly used in complexity theory, is the Exponential Time Hypothesis by Impagliazzo and Paturi [27], which implies that 3-SAT with n variables cannot be solved in time \(2^{o(n)}\). For example, assuming the ETH, 3-COLORING, INDEPENDENT SET, DOMINATING SET, VERTEX COVER, HAMILTONIAN CYCLE cannot be solved in time \(2^{o(n)}\) in general graphs or in time \(2^{o(\sqrt{n})}\) in planar graphs. Thus the algorithms we know are asymptotically tight, unless the ETH fails.

A natural direction of research is to consider restricted graph classes and try to classify problems that can be solved in subexponential time. Geometric intersection graphs provide a rich family of graph

\[^1\]In the \(\tilde{O}(\cdot)\) notation we suppress polylogarithmic factors
classes, which are potentially interesting from the point of view of fine-grained complexity, as they lie “in between” planar graphs and all graphs. For a family $S$ of sets, we define its intersection graph, whose vertices are in one-to-one correspondence to members of $S$, and two vertices are adjacent if and only if their corresponding sets intersect. We will be interested in intersection graphs of sets of geometric objects in the plane.

For example, in unit disk graphs, i.e., intersection graphs of unit-radius disks in the plane, Independent Set, Hamiltonian Cycle, Vertex Cover can be solved in time $2^{O(\sqrt{n})}$ \cite{11, 15, 38, 39}, and $k$-Coloring can be solved in time $2^{O(\sqrt{k}\log k)}$ for every $k \geq 3$ \cite{28}. All these bounds are essentially tight under the ETH, up to a polylogarithmic factors in the exponent. Many of algorithms for (unit) disk graphs use the fact, that disk intersection graphs also have small separators. Indeed, Miller et al. showed that the intersection graph of a family of $n$ disks, such that at most $k$ of them share a single point, has a balanced separator of size $O(\sqrt{nk})$ \cite{38}. This was later generalized to intersection graphs of families of arbitrary convex shapes that are fat, i.e., with bounded ratio of the radius of the smallest enclosing circle to the radius of the largest enclosing circle \cite{37, 38}.

It is perhaps interesting to note that, by the celebrated kissing lemma by Koebe \cite{29}, every planar graph is an intersection graph of interior-disjoint disks. Note that in such a representation each point is contained in at most two disks, so the separator theorem for disk graphs implies the planar separator theorem (up to constants).

In this paper we will be more interested in intersection graphs of non-fat geometric objects. In particular, we will investigate string graphs, which are intersection graphs of continuous curves in the plane (see Kratochvíl \cite{39, 31} and segment graphs, i.e., intersection graphs of straight-line segments (see Kratochvíl and Matoušek \cite{33}). We can restrict the representation even further and consider $k$-DIR graphs, which are intersection graphs of segments using at most $k$ distinct slopes \cite{33}. It is worth to mention that planar graphs form another subclass of segment graphs \cite{6, 18}.

More and more refined string separator theorems have been proven by Fox and Pach \cite{16} and Matoušek \cite{34}. Finally, Lee \cite{34} proved the following, asymptotically tight, version.

**Theorem 1** (Lee \cite{34}). Every string graph with $m$ edges has a balanced separator of size $O(\sqrt{m})$. It can be found in polynomial time, if the geometric representation is given.

Observe that since planar graphs are string graphs and have linear number of edges, Theorem 1 implies planar separator theorem (again, up to constants).

Using the string separator theorem, Fox and Pach \cite{17} showed that Independent Set (and thus Vertex Cover) can be solved in subexponential time in string graphs. Combining Theorem 1 with their approach gives the complexity $2^{O(n^{2/3})}$. The algorithm is a simple win-win strategy: either we have a vertex of large degree and we branch on choosing it to the solution or not, or all degrees are small and thus there exists a small balanced separator, which allows us for one step of divide & conquer. Recently, Marx and Pilipczuk \cite{35, 39} used a different approach to obtain a $2^{O(\sqrt{n})}p^{O(1)}$ algorithm for Independent Set in string graphs, where $p$ is the number of geometric vertices in the representation.

While the algorithm of Marx and Pilipczuk seems difficult to generalize to other problems, Bonnet and Razgowski \cite{3, 4} showed that the win-win strategy of Fox and Pach can be successfully applied to obtain subexponential algorithms for 3-Coloring, Feedback Vertex Set and Max Induced Matching.

Quite surprisingly, they showed that for every $k \geq 4$, $k$-Coloring cannot be solved in time $2^{o(n)}$, even in 2-DIR graphs, unless the ETH fails. They also showed that assuming the ETH, Dominating Set, Independent Dominating Set, and Connected Dominating Set do not admit subexponential algorithms in segment graphs, and CLIQUE does not admit such algorithm in string graphs.

This shows that the complexity landscape in string and segment graphs appears to be much more interesting than in planar graphs or intersection graphs of fat objects. In order to understand which problems can be solved in subexponential time, it would be especially desirable to obtain full dichotomy theorems for some natural families of problems, instead of proving ad-hoc results for single problems. A natural language to describe families of problems in a uniform way is provided by graph homomorphisms. For two graphs $G$ and $H$ (with possible loops), a homomorphism from $G$ to $H$ is an edge-preserving mapping from the vertex set of $G$ to the vertex set of $H$ (see the comprehensive book by Hell and Nešetřil \cite{24}). Note that the notion of homomorphisms is flexible and allows us to impose additional restrictions, such as vertex/edge lists \cite{10, 25} or vertex/edge weights \cite{21}. This way many well-known problems can be equivalently formulated as problems of finding a homomorphism to certain graph $H$, possibly with additional constraints. For example, $k$-coloring is equivalent to a homomorphism to $K_k$, and Independent Set is equivalent to a weight-maximizing homomorphism to $H = \begin{array}{c} \circ \end{array} \begin{array}{c} \circ \end{array} \begin{array}{c} \circ \end{array}$ (numbers denote weights assigned to vertices of $G$ assigned to particular vertices of $H$).
**Weighted homomorphisms.** Let $H = (V(H), E(H))$ be a fixed graph (with possible loops), and consider the following computational problem called $\text{WHom}(H)$. The instance consists of a graph $G = (V(G), E(G))$, a weight function $w: (V(G) \times V(H)) \cup (E(G) \times E(H)) \rightarrow \mathbb{R}$, and an integer $k$. For simplicity we will also allow $-\infty$ as a weight, but this can be avoided by shifting all weights and using a sufficiently small integer to represent the weight corresponding to a forbidden choice. For a homomorphism $h: G \rightarrow H$, we define its weight as $\sum_{v \in V(G)} w(v, h(v)) + \sum_{e \in E(G)} w(e, h(e))$. We ask whether there exists a homomorphism from $G$ to $H$, whose total weight is at least $k$. It is straightforward to see that this problem generalizes some well-studied variants of graph homomorphism problem, including $\text{LIST HOMOMORPHISM}$ [10] and $\text{MIN COST HOMOMORPHISM}$ [21].

We show the following dichotomy theorem for $\text{WHom}(H)$ in string graphs.

**Theorem 2.** Let $H$ be a fixed graph.
(a) If $H$ has no two vertices with a common neighbor, then the $\text{WHom}(H)$ problem can be solved in time $2^{O(n^{2/3} \log n)}$ for string graphs with $n$ vertices.
(b) Otherwise, there is no algorithm solving the $\text{WHom}(H)$ problem in time $2^{o(n)}$ for segment graphs with $n$ vertices, unless the ETH fails.

Interestingly, very recently Groenland et al. [21] observed that if $H$ has no two vertices with two common neighbors, then $\text{WHom}(H)$ can be solved in time $2^{O(\sqrt{n})}$ in $P_t$-free graphs, for every fixed $t$. Note that there are string graphs with arbitrarily long induced paths, and there are $P_t$-free graphs that are not string graphs.

The algorithm proving part (a) of Theorem 2 is a slight adaptation of the win-win approach by Fox and Pach [17] later extended by Bonnet and Rzążewski [13, 14] and Groenland et al. [20]. The proof of part (b) is divided into several cases, depending on the structure of $H$. In our constructions we always try not to use the whole expressibility of the $\text{WHom}(H)$ problem, but to obtain the result even for some natural special cases.

All hardness proofs follow the same pattern – we start with a grid-like arrangement of segments, inducing a clique or a biclique, and then add constant-size gadgets to encode a specific problem. Note that this requires the objects to be non-fat and gives some intuition why problems in segment graphs tend to be harder than in intersection graphs of fat objects, and how the hardest instances look like.

Finally, all graphs we construct are actually $P_t$-free for some fixed $t$, which, along with the result of Groenland et al. [20], gives the following dichotomy theorem for $P_t$-free graphs.

**Theorem 3.** Let $H$ be a fixed graph.
(a) If $H$ has no two vertices with a common neighbor, then for all fixed $t$ the $\text{WHom}(H)$ problem can be solved in time $2^{O(\sqrt{n})}$ for $P_t$-free graphs with $n$ vertices.
(b) Otherwise, there is no algorithm solving the $\text{WHom}(H)$ problem in time $2^{o(n)}$ for $P_t$-free graphs with $n$ vertices, unless the ETH fails.

**Locally constrained homomorphisms.** Interesting variants of graph homomorphism problems can be obtained by imposing some additional constrains on the neighborhood of each vertex. A homomorphism $h$ from $G$ to $H$ is called locally injective (locally bijective, locally surjective) if for every $v \in V(G)$ it induces an injective (bijective, surjective, resp.) mapping between the neighborhood of $v$ and the neighborhood of $f(v)$. Locally bijective homomorphisms have been studied from combinatorial [12, 14] and computational point of view [7, 11, 13, 37]. Let $\text{LIHom}(H)$, $\text{LBHom}(H)$, and $\text{LSHom}(H)$ denote, respectively, the computational problems of determining the existence of a locally injective, bijective, and surjective homomorphism from a given graph to $H$.

Some well-known graph problems can be expressed as locally constrained homomorphism. For example, locally injective homomorphism to the complement of the $k$-vertex path appears to be equivalent to $k$-$L(2, 1)$-labeling, i.e., a mapping from the vertex set of the input graph to the set $\{1, 2, \ldots, k\}$, in which adjacent vertices get labels differing by at least 2, and vertices with a common neighbor get different labels [10, 23]. If $H$ is a complete graph $K_k$, then $\text{LIHom}(H)$ is exactly the $k$-coloring of the square of the graph [36, 49]. Finally, if $H$ is a complete graph with $k$ vertices, where every vertex has a loop, then $\text{LIHom}(H)$ is equivalent to the so-called injective $k$-coloring [22, 20], in which the only restriction is that no two vertices with a common neighbor get the same color.

We show that, unlike $\text{WHom}(H)$, both $\text{LIHom}(H)$ and $\text{LBHom}(H)$ can always be solved in subexponential time in string graphs.

**Theorem 4.** For every fixed graph $H$, the $\text{LIHom}(H)$ problem and the $\text{LBHom}(H)$ problem can be solved in time $2^{O(\sqrt{n \log n})}$ in string graphs with $n$ vertices.
The \text{LSHom}(H) problem appears to be much harder. In particular, we show the following dichotomy for simple graphs $H$ with $\Delta(H) \leq 2$ (observe that if $|H| \leq 2$, the problem can trivially be solved in polynomial time).

**Theorem 5.** Let $H$ be a connected simple graph with $\Delta(H) \leq 2$ and $|H| \geq 3$. 
(a) If $H \in \{P_3, C_4\}$, then the \text{LSHom}(H) problem can be solved in time $2^{O(n^{2/3}\log^{2/3}n)}$ for string graphs on $n$ vertices, even if geometric representation is not given.
(b) Otherwise, the \text{LSHom}(H) problem cannot be solved in time $2^{o(n)}$ in 2-DIR graphs with $n$ vertices, unless the ETH fails.

We also show that \text{LSHom}(H) cannot be solved in subexponential time for $H = \infty \circ \circ$. Note that none of the graphs $H$, for which we consider the \text{LSHom}(H) problem, has two vertices with two common neighbors. Thus they are all “easy” cases of the \text{WHom}(H).

**Representation and robust algorithms.** When dealing with geometric intersection graphs, we need to be careful, whether the input consist of the graph along with the representation, or just the graph (with a a promise that a geometric representation exists). This distinction might be crucial, since finding the representation is often a computationally hard task.

Recognizing string and segment graphs was shown to be NP-hard by Kratochvíl \cite{Kra} and Kratochvíl and Matoušek \cite{KM}, respectively. However, for a very long time it was unclear whether these problems are in NP. This is because there are string graphs, whose every representation requires exponential number of crossing points \cite{Ber} and there are segment graphs, whose every representation requires points with double exponential coordinates \cite{Kra, Mat}. Finally, Schaefer, Sedgwick, and Stefankovič showed that recognizing string graph is in NP \cite{Sch}, while recognizing segment graph appears to be complete for the complexity class $\exists \forall$ \cite{Hes, Hes2}. This is a strong indication that the problem might not be in NP.

For these reasons, it is if an algorithm does not require explicit representation. Such algorithms are called robust – they either compute the solution, or report that the input graph does not belong to the required class. All algorithms presented in the paper are robust, but can be made slightly faster, if the representation is given. On the other hand, all hardness results hold even if the graph is given along with the geometric representation.

## 2 Weighted homomorphism problem

In this section we prove \textbf{Theorem 2}. Let us first discuss the notation and some preliminary observations. For every vertex $v$ of graph $G$, let $N_G(v)$ denote the set of neighbors of $v$ in $G$ and for any $V' \subseteq V(G)$ let $N_G(V') = \bigcup_{v \in V} N_G(v)$. Let $\deg_G(v) = |N_G(v)|$. If the graph is clear from the context, we will omit the subscript $G$ and simply write $N(v)$ and $\deg(v)$.

By $\Delta(G)$ we denote $\max_{v \in V(G)} \deg(v)$.

Let us recall the definition of $\text{WHom}(H)$. The instance consists of a graph $G$, an integer $k$, and a \textit{weight function} $w : (V(G) \times V(H)) \cup (E(G) \times E(H)) \rightarrow \mathbb{R}$. For a homomorphism $h : G \rightarrow H$ and for any $X \subseteq V(G) \cup E(G)$ we define the weight of $X$ by $w_h(X) = \sum_{x \in X} w(x, h(x))$. The weight of $h$ is defined as $w_h(V(G) \cup E(G))$. We ask if there exists a homomorphism, whose weight is at least $k$. A homomorphism $h : G \rightarrow H$ will be often called a \textit{coloring} of $G$ and we will think of vertices of $H$ as colors.

We will say that $H$ (with possible loops) has \textit{property} $(\ast)$, if it does not contain any pair of vertices $u, v$ such that $|N(u) \cap N(v)| \geq 2$. Note that if $H$ is loopless, then it has property $(\ast)$ iff it does not have a subgraph isomorphic to $C_4$.

### 2.1 Algorithm

In this section we prove statement (a) of \textbf{Theorem 2}

**Theorem 2 (a).** Let $H$ be a fixed graph. If $H$ has no two vertices with a common neighbor, then the \text{WHom}(H) problem can be solved in time $2^{O(n^{2/3}\log^{2/3}n)}$ for string graphs with $n$ vertices.

**Proof.** Let $G$ be a graph and $w$ be the weight function. We will find a homomorphism from $G$ to $H$, maximizing the total weight (if one exists). Actually, we will assume that we are additionally given lists $L : V(G) \rightarrow 2^{V(H)}$, and we ask for a homomorphism respecting these lists. Note that this is not really necessary, as list can be expressed with appropriate choice of weights, but this requires modifying the...
weight function and makes the argument more complicated. Define \( N := \sum_{v \in V(G)} |L(v)| \). Let \( \ell := 2^{|H|} \), note that there are at most \( \ell \) possible lists \( L(v) \).

We start with preprocessing the instance. For any two adjacent vertices \( u \) and \( v \) of \( G \), if \( L(v) \) contains a vertex \( b \in V(H) \), which is non-adjacent to every vertex from \( L(u) \), we can safely remove \( v \) from \( L(u) \). Moreover, if there exists a vertex \( v \in V(G) \), such that \( L(v) \) is a singleton, say \( L(v) = \{ a \} \), then we can map \( v \) to \( a \) and remove it from \( G \). We repeat these steps while possible, this clearly takes only polynomial time, as \( N \leq n |H| \). Due to this step, we can assume that every list has at least two elements and for every \( uw \in E(G) \) and every \( a \in L(v) \) there exists \( a' \in L(u) \) such that \( aa' \in E(H) \).

First, consider the case that \( G \) has a vertex \( v \) such that \( \deg(v) > n^{1/3} \). It means that there is a list \( L \), which is assigned to at least \( n^{1/3}/\ell \) neighbors of \( v \). We observe that there exist \( a \in L(v) \) and \( b \in L \) such that \( ab \notin E(H) \). Indeed, due to the preprocessing step, \( |L(v)| \geq 2 \) and \( |L| \geq 2 \). So if every element of \( L(v) \) is adjacent to every element of \( L \), then the property \((\ast)\) is violated.

We branch on assigning \( a \) to \( v \): either we remove \( a \) from \( L(v) \) or we color \( v \) with \( a \) and update the lists of neighbors of \( v \) by removing from them every non-neighbor of \( a \). In particular, we will remove \( b \) from lists of at least \( n^{1/3}/\ell \) neighbors of \( v \). Let \( F(N) \) be the complexity of this step, clearly is it bounded by the following:

\[
F(N) \leq F(N - 1) + F(N - n^{1/3}/\ell) \leq \left( n^{1/3}/\ell + 1 \right)^{N/\ell n^{1/3}} \leq 2^{O(n \log n / n^{1/3})} \leq 2^{O(n^{2/3} \log n)}.
\]

In the other case we have that every vertex of \( G \) has degree at most \( n^{1/3} \). This means that \( G \) has \( O(n^{1/3}) \) edges, so, by Theorem 1 there exists a balanced separator \( S \) of size \( O(n^{2/3}) \). We can find \( S \) in polynomial time using the geometric representation, or by exhaustive guessing in time \( nO(n^{2/3}) = 2^{O(n^{2/3} \log n)} \), if the geometric representation is not given. Then, we consider all possible \( H \)-colorings of \( S \) and run one step of a standard divide and conquer algorithm. The complexity of this step is \( nO(n^{2/3}) = 2^{O(n^{2/3} \log n)} \) and so is the total combined complexity of the algorithm.

Observe that even if the input graph \( G \) is not a string graph, but has an appropriate structure of separators, the algorithm can still give the correct answer in time \( 2^{O(n^{2/3} \log n)} \). Otherwise, the exhaustive search for a separator will fail and we can report that \( G \) is not a string graph. This means that the presented algorithm is robust. Moreover, the algorithm can be easily adapted to count all feasible solutions.

### 2.2 Hardness results

Now we show that property \((\ast)\) is essential for the existence of subexponential algorithms: for all remaining graphs \( H \), an algorithm solving \( WHom(H) \) for string graphs in subexponential time would contradict the Exponential Time Hypothesis. To begin with, observe that we can express the property \((\ast)\) in terms of forbidden subgraphs.

**Observation 6.** A graph has property \((\ast)\) if and only if it does not contain any of the seven graphs shown on the Figure 1 as an induced subgraph.

![Figure 1: Characterization of property (\ast) by forbidden induced subgraphs.](image)

Note that to prove Theorem 2 (b), it is enough to show hardness of \( WHom(H) \) for graphs \( H \) shown in Fig. 1. Let \( H \) be an induced subgraph of \( H' \). Consider an instance \((G, w', k)\) of \( WHom(H) \). Define \( w' : V(G) \times V(H') \cup E(G) \times E(H') \rightarrow \mathbb{R} \) as follows: for \( x \in V(G) \cup E(G) \), if \( a \in V(H') \, \cup \, E(H') \), then \( w'(x, a) = w(x, a) \), otherwise \( w'(x, a) = -\infty \). Note that \((G, w', k)\) is an instance of \( WHom(H') \), equivalent to the instance \((G, w, k)\) of \( WHom(H) \).

We prove Theorem 2 (b) for the graph (a) in Section 2.2.1 for (b) in Section 2.2.2 and for all the remaining cases in Section 2.2.3. Note that the problem problem of finding a homomorphism to \( K_4 \) (the graph \( (G) \)) is exactly 4-COLORING. It is known that assuming the ETH, this problem does not admit a subexponential algorithm, even for 2-DIR graphs [H 3].
2.2.1 Maximum Cut

In this section, $H$ is the graph (a) from Fig. 1. Note that any function $h: V(G) \rightarrow V(H)$ is a homomorphism and thus determining the existence of a homomorphism with vertex lists and weights is trivial. However, it becomes more interesting if we include the edge weights.

We denote the vertices of $H$ by $a$ and $b$, so we have $E(H) = \{aa, ab, bb\}$ (see Fig. 2). We also define the weight function as follows. Let $w(v, u) = 0$ for every $(v, u) \in V(G) \times V(H)$, and for every $e \in E(G)$ we set $w(e, aa) = w(e, bb) = 0$ and $w(e, ab) = 1$. Note that the value of $w(e, f)$ does not depend on $e$, so $w$ is in fact an edge-weighting of $H$ (see Fig. 2).

$$\begin{array}{c|c|c}
0 & 1 & 0 \\
\hline
a & & b \\
\end{array}$$

Figure 2: Graph $H$ with its corresponding edge weights defined by $w$.

Observe that the weight of a homomorphism $h: G \rightarrow H$ equals the number of edges mapped to $ab$. Thus finding a homomorphism of maximum weight is equivalent to partitioning the $V(G)$ into two subsets, so that the number of edges crossing this partition is maximized. Such a set of edges is called a cut in $G$ and the computational problem of finding the maximum cut is denoted by MAX CUT. We say that a cut $(A, B)$ of $G$ is a bisection, if $|A| = |B|$. We will use the following lemma, which says that MAX CUT remains hard for graphs $G$ of bounded degree, even if every maximum cut in $G$ is a bisection.

**Lemma 7.** Assuming the ETH, there is no algorithm solving MAX CUT in time $2^{o(n)}$ for a graph $G$ on $n$ vertices, even if $\Delta(G) \leq 7$ and every maximum cut in $G$ is a bisection.

To prove Lemma 7, we present a sequence of linear reductions, starting from a well-known problem PosNAE 3-Sat. In PosNAE 3-Sat, for a given set of variables $U$ and clauses $C$, such that every clause contains at most 3 variables and all of them are non-negated, we ask if there exists a truth assignment $f: U \rightarrow \{0, 1\}$ such that for every clause $c \in C$ we have that $f(c) = \{0, 1\}$. It is known that, if we assume the ETH, there is no algorithm to solve PosNAE 3-Sat on $n$ variables in time $2^{o(n)}$, even if each variable occurs at most 3 times. We start with the following lemma.

**Lemma 8.** There is no algorithm solving MAX CUT in time $2^{o(n)}$ for graphs $G$ on $n$ vertices, even if $\Delta(G) \leq 6$, unless the ETH fails.

**Proof.** Let $\Phi = (U, C)$ be a PosNAE 3-Sat formula such that $U = \{u_1, \ldots, u_n\}$ and $C = \{c_1, \ldots, c_m\}$. We can assume that all clauses are distinct and each clause $c_i$ contains 2 or 3 variables, as if $|c_i| = 1$, then clearly the satisfying assignment does not exist. Let $m_2$ and $m_3$, respectively, be the number of clauses of each cardinality.

For $\Phi$, we construct a graph $G$, in which $\Delta(G) \leq 6$ and $|V(G)| = O(n)$. For every variable $u_i$ we create a variable vertex $x_i$. For every 2-element clause $c = \{u_i, u_j\}$, we add an edge $x_i x_j$. For every 3-element clause $c = \{u_i, u_j, u_k\}$, we add a gadget containing six new vertices $l_1, r_1, l_2, r_2, l_3, r_3$ and nine new edges in a way that $(l_1, x_i, r_1, l_2, x_j, r_2, l_3, x_k, r_3)$ induce a 9-cycle (see Figure 3 (a)).

Let us now show that $\Phi$ is satisfiable if and only if $G$ has a cut of size at least $m_2 + 8m_3$. First, assume that $\Phi$ is satisfiable. Let $f$ be a satisfying assignment and let us define the cut $(A, B)$ of $G$. For every $i$, we set $x_i \in A$ iff $f(u_i) = 0$. Since $f$ is a satisfying assignment, all edges coming from 2-element clauses are in the cut. If $v$ is not a variable vertex, observe that it is adjacent to exactly one variable vertex $x_i$, and we set $v \in A$ iff $f(u_i) = 1$. This way, since $f$ is satisfying, in each clause gadget we have eight edges in the cut, which gives $m_2 + 8m_3$ in total (see Figure 3 (b)).

Now let $(A, B)$ be a cut in $G$ of size at least $m_2 + 8m_3$. This means it has size exactly $m_2 + 8m_3$, as $|E(G)| = m_2 + 9m_3$ and there are $m_3$ edge-disjoint cycles $C_3$ in $G$. Moreover, exactly 8 edges from each clause gadget belong to the cut, so exactly two adjacent vertices are in the same part of cut. This gives us that for every clause at least one of its variable vertices is in $A$ and at least one is in $B$ (see Figure 3 (b)). Also, as the cut has size $m_2 + 8m_3$, all edges from 2-element clause gadgets must belong to the cut as well. So the truth assignment $f: U \rightarrow \{0, 1\}$ such that $f(u_i) = 0$ iff $x_i \in A$ satisfies $\Phi$.

Finally, we observe that as each variable occurred in at most 3 clauses, we have $m \leq 3n$, so $|V(G)| \leq n + 6m = O(n)$, which completes the proof.

Now we are ready to prove Lemma 7.
Proof of Lemma 8. Let \((G, k)\) be an instance of \textsc{Max Cut}, such that \(\Delta(G) \leq 6\) and \(|V(G)| = n\). Construct a graph \(F\) as follows: take a copy of \(G\), denoted by \(G'\), and for each its vertex \(v\) add a vertex \(v'\) and an edge \(vv'\). Clearly, \(|V(F)| = 2n\) and \(\Delta(F) \leq 7\).

First, we observe that if \((A_F, B_F)\) is a maximum cut of \(F\), then it is a bisection. Assume the opposite, i.e., \((A_F, B_F)\) is a maximum cut such that \(|A_F| < |B_F|\). As \(|V(G')| = |V(F) \setminus V(G')|\), there exists \(v \in V(G')\), such that \(v, v' \in B_F\). Note that \((A_F \cup \{v'\}, B_F \setminus \{v'\})\) is a cut of larger size, a contradiction.

Now let us show that graph \(G\) has a cut of size at least \(k\) if and only if there exists a bisection in \(F\) of size at least \(n + k\). Assume that \((A, B)\) is a cut in \(G\) of size at least \(k\). Let \((A_F, B_F)\) be a cut in \(F\) such that every vertex \(v\) of \(G'\) is in \(A_F\) iff \(v \in A\), and every vertex \(v'\) is in \(V(F) \setminus V(G')\) iff its neighbor is in \(B\). Note that \((A_F, B_F)\) is a bisection, because \(|A_F| = |B_F| = |A| + |B|\). There are at least \(k\) edges between \(A_F \cap V(G')\) and \(B_F \cap V(G')\) and another \(n\) edges between \(V(G')\) and \(V(F) \setminus V(G')\), so \((A_F, B_F)\) is a bisection of \(F\) of size at least \(n + k\). For the converse, assume that \((A_F, B_F)\) is a bisection in \(F\) of size at least \(n + k\). Observe that at most \(n\) edges in \(F\) do not belong to \(E(G')\), which means \((A_F \cap V(G'), B_F \cap V(G'))\) is a cut of \(G\) of size at least \(k\).

This, combined with Lemma 8, completes the proof.

For our result, it is enough to show the hardness of \textsc{Max Cut} on segment graphs.

**Theorem 9.** There is no algorithm solving \textsc{Max Cut} for segment graphs \(G\) on \(n\) vertices in the time \(2^{o(n)}\), unless the ETH fails.

**Proof.** Let \((G, k)\) be a given instance of \textsc{Max Bisection}, such that \(\Delta(G) \leq 7\) and every maximum cut in \(G\) is a bisection. Let \(V(G) = \{v_1, \ldots, v_n\}\) and \(m := |E(G)|\). We provide an instance \(G^*\) of \textsc{Max Cut}, where \(G^*\) is a segment graph, such that \(G\) has a bisection of size at least \(k\) if and only if \(G^*\) has a cut of size at least \(n^2 + 32n + 4m + 2k\), which equivalently means that there exists a homomorphism from \(G\) to \(H\) of weight at least \(n^2 + 32n + 4m + 2k\).

We start constructing \(G^*\) with two sets \(X = \{x_1, \ldots, x_n\}\) and \(Y = \{y_1, \ldots, y_n\}\) of segments. The segments of each set intersect in a single point, and all segments altogether are arranged in a grid-like manner (see Figure 4).

![Figure 4: First step of the construction of \(G^*\).](image)

For each vertex \(v_i\) of \(G\), we add a **vertex gadget** on the intersection of \(x_i\) and \(y_i\). It contains a set \(D_i\) of 16 parallel non-overlapping segments, crossing both \(x_i\) and \(y_i\) (see Figure 5). For each edge \(v_iv_j\) of \(G\) we define an **edge gadget** on the intersection of \(x_i\) and \(y_j\) by putting two additional segments \(\alpha_{ij}\) and \(\beta_{ij}\), crossing each other and, respectively, \(x_i\) or \(y_j\). Note that each edge of \(v_iv_j\) of \(G\) is represented by two edge gadgets, one on the intersection of \(x_i\) and \(y_j\), and another one on the intersection of \(x_j\) and \(y_i\). Define \(E_i := \bigcup_{v_j \in N(v_i)} \{\alpha_{ij}, \beta_{ij}\}\).

We say a homomorphism \(h : G^* \rightarrow H\) is **pretty**, if the following properties are satisfied:

- **P1.** \(|h(D_i)| = 1\) for every \(v_i \in V(G)\),
- **P2.** \(h(x_i) = h(y_i)\) and \(h(x_i) \not\in h(D_i)\) for every \(v_i \in V(G)\),
P3. \( h(\alpha_{ij}) \neq h(x_i) \) and \( h(\beta_{ij}) \neq h(y_j) \) for every \( v_i v_j \in E(G) \).

**Claim 1.** If there exist a homomorphism \( h : G^* \to H \) of weight \( p \), then there exists a pretty homomorphism from \( G^* \) to \( H \) of weight at least \( p \).

**Proof of Claim.** Consider a homomorphism \( h : G^* \to H \). Note that recoloring all segments in \( D_i \) to the color other than \( h(x_i) \) does not decrease the weight, so we can assume that P1 holds for \( h \). So suppose that P2 does not hold and let \( v \) be a vertex for which \( h(x_i) \neq h(y_j) \), without loss of generality \( h(x_i) = a \).

Let \( A \) (resp., \( B \)) be the set of segments from \( (X \cup Y) \setminus \{x_i, y_j\} \), that are mapped to \( a \) (resp., \( b \)).

**Case 1:** If \( |A| \geq |B| \), then consider a homomorphism \( h' : G^* \to H \), obtained from \( h \) by recoloring \( x_i \) to \( b \) and all segments in \( D_i \) to \( a \). We will show that \( w_h(E(G^*))) > w_{h'}(E(G^*)) \). Clearly, the weight can differ only on edges which have at least one endpoint in the set \( D_i \cup \{x_i, y_j\} \), denote the set of these edges by \( E' \). First, let us count the edges from \( E' \) that were mapped to \( ab \) by \( h \), i.e., that contribute to \( w_h(E(G^*)) \). There are \( |A| + |B| + 1 \) such edges with both endpoints in \( X \cup Y \), 16 with one endpoint in \( D_i \), and at most 14 with one endpoint in \( E_i \), which gives at most \( |A| + |B| + 31 \) in total. Now let us count the edges from \( E' \) that are mapped to \( ab \) by \( h' \). There are \( 2|A| \) such edges with both endpoints in \( X \cup Y \), 32 with one endpoint in \( D_i \) and some number (possibly zero) of edges with one endpoint in \( E_i \).

Observe that \( w_h(E(G^*))) - w_{h'}(E(G^*)) \geq (2|A| + 32) - (|A| + |B| + 31) > 0 \).

**Case 2:** If \( |A| < |B| \), then we consider a homomorphism \( h' : G^* \to H \), obtained from \( h \) by recoloring \( y_j \) to \( a \) and all segments in \( D_i \) to \( b \). With the reasoning analogous to the one in Case 1, we show that the weight of \( h' \) is at least the weight of \( h \).

After at most \( n \) such recolorings, we receive a homomorphism satisfying P1 and P2, so now we will assume that P1 and P2 hold in \( h \). Suppose that there is \( v_i v_j \) violating P3. Again, we recolor some segments to obtain a new homomorphism \( h' \). If \( h(x_i) = h(y_j) \), then we recolor \( \alpha_{ij} \) and \( \beta_{ij} \) to the color other than \( h(y_j) \). If \( h(x_i) \neq h(y_j) \), then we color \( \alpha_{ij} \) with the color \( h(y_j) \), and \( \beta_{ij} \) with the color \( h(x_i) \).

Observe that \( w_h(E(G^*))) \) is not lower than the weight of \( h \) and \( h' \) still satisfies P1 and P2. Repeat this for every edge which does not satisfy P3. This way, after at most \( 2m \) steps, we obtain a pretty homomorphism.

Now let us show that \( G \) has a bisection of size at least \( k \) if and only if there exists homomorphism from \( G^* \) to \( H \) of weight at least \( n^2 + 32n + 4m + 2k \), i.e., a cut in \( G^* \) of size at least \( n^2 + 32n + 4m + 2k \).

First assume there exists a bisection \( (A_G, B_G) \) of \( G \) of size at least \( k \). Let us define a homomorphism \( h : G^* \to H \). For every \( v_i \in A_G \), we set \( h(x_i) = h(y_i) = a \), and for every \( v_i \in B_G \), we set \( h(x_i) = h(y_i) = b \). Moreover, for every \( i \), we color all segments from \( D_i \) with the color \( h(x_i) \). Finally, note that every \( \alpha_{ij} \) and every \( \beta_{ij} \) has exactly one neighbor in \( X \cup Y \), and we color it with the color other than this neighbor.

Let us count edges mapped to \( ab \). Clearly, there are \( n^2 \) such edges with both endpoints in \( X \cup Y \), and another \( 32n \) in vertex gadgets. For every \( i, j \), such that \( v_i v_j \) is in the cut \( (A_G, B_G) \), we get another three edges mapped to \( ab \) from the edge gadget, which gives \( 6k \) in total (recall that each edge is represented twice). If \( v_i v_j \) is not in the cut, then only two edges from each edge gadget are mapped to \( ab \), which gives \( 4(m - k) \) edges in total. Summing up, the weight of \( h \) is \( n^2 + 32n + 4m + 2k \).

Now let \( h : G^* \to H \) be a homomorphism of weight at least \( n^2 + 32n + 4m + 2k \). According to Claim 1, we can assume that \( h \) is pretty. Define a partition \( (A_G, B_G) \) of \( V(G) \) in the following way: if \( h(x_i) = a \), then \( v_i \in A_G \) and if \( h(x_i) = b \), then \( v_i \in B_G \). Let \( k' \) be the number of edges between the sets \( A_G \) and \( B_G \).

As \( h \) is pretty, we can easily count the edges mapped to \( ab \). There are \( n^2 \) such edges between the vertices of \( X \cup Y \), \( 32n \) edges with one endpoint in \( \bigcup_{i=1}^{n} D_i \), and \( 4m + 2k' \) edges with at least one endpoint in \( \bigcup_{i=1}^{n} E_i \). So we have that \( n^2 + 32n + 4m + 2k' \geq n^2 + 32n + 4m + 2k \), which implies that \( k' \geq k \). This means that there exists a cut in \( G \) of size at least \( k \). As we know that each maximum cut of \( G \) is a bisection, we know that there also exists a bisection of \( G \) of size at least \( k \).
To complete the proof, we observe that $G^*$ has $18n + 4m \leq 30n = O(n)$ vertices.

## 2.2.2 Minimum odd-cycle transversal

Now let us consider the case when $H$ is the the graph (b) in [Figure 6](#). This time we will consider a vertex-weighted variant. Denote the vertices of $H$ by $a, b, c$, where $a$ is the vertex with the loop. All edge-weights are set to 0. For vertex weights, for every $v \in V(G)$ we set $w(v, b) = w(v, c) = 1$ and $w(v, a) = 0$. Note that again the weights do not depend on the choice of $v$, so we can think of $w$ as a vertex-weighting of $H$ (see Figure 6).

![Figure 6](#) The graph $H$ with its corresponding vertex weights defined by $w$.

We observe that finding a homomorphism $h$ of maximum weight is equivalent to the problem of finding the maximum number of vertices of $G$ which induce a bipartite subgraph, or, equivalently, the minimum number of vertices, whose removal destroys all odd cycles. This problem is known as Odd Cycle Transversal. We aim to show the following.

**Theorem 10.** The Odd Cycle Transversal problem in 2-DIR graphs with $n$ vertices cannot be solved in time $2^{o(n)}$, unless the ETH fails.

**Proof.** This time we will reduce from the Independent Set problem, which cannot be solved in time $2^{o(n)}$, even if the input graph has a bounded maximum degree. Let $(G, k)$ be an instance of Independent Set, such that $V(G) = \{v_1, \ldots, v_m\}$, $|E(G)| = m = O(n)$. We will construct a segment graph $G^*$, which admits a homomorphism to $H$ with weight at least $7n + 2m + k$ if and only if $G$ has an independent set of size at least $k$.

We start the construction of $G^*$ with introducing the set $X = \{x_1, \ldots, x_n\}$ of disjoint, parallel, horizontal segments and the set $Y = \{y_1, \ldots, y_n\}$ of vertical segments, such that $X$ and $Y$ form a grid. For every vertex $v_i$, on the intersection of $x_i$ and $y_i$ we add a vertex gadget shown on Figure 7(a). Let $D_i = \{d_1, \ldots, d_7\}$. For every $i, j$, such that $v_i v_j \in E(G)$, on the intersection of $x_i$ and $y_j$ we add an edge gadget shown on Figure 7(b). Note that again each edge is represented by two edge gadgets.

![Figure 7](#) (a) A vertex gadget for $v_i \in V(G)$ and (b) an edge gadget for $v_i v_j \in E(G)$.

We say that a homomorphism $h: G^* \to H$ is pretty, if

- **P1.** $h(e_{ij}) \in \{b, c\}$ for every $v_i v_j \in E(G)$,
- **P2.** $h(x_i) = a$ iff $h(y_i) = a$ for every $v_i \in V(G)$,
- **P3.** if $h(x_i) = h(y_i) = a$, then $w_h(D_i) = 7$, otherwise $w_h(D_i) = 6$.

**Claim 2.** Let $p > 7n + 2m$. If there exist a homomorphism $h: G^* \to H$ of weight $p$, then there exists a pretty homomorphism from $G^*$ to $H$ of weight at least $p$.

**Proof of Claim.** First, observe that it is impossible that $\{b, c\} \subseteq h(X)$. Indeed, in such a case we would have $h(Y) = \{a\}$, and for every $v_i \in V(G)$ such that $h(x_i) \in \{b, c\}$, the total weight of $D_i$ is at most 6. Since the total weight of all segments in edge gadgets is at most $2m$, we obtain that $p \leq n + 6n + 2m$, a contradiction. Analogously we can show that $\{b, c\} \not\subseteq h(Y)$. Thus, by symmetry, we may assume that $h(X) \subseteq \{a, b\}$ and $h(Y) \subseteq \{a, c\}$.

Assume that P1 is not satisfied. Let $E'$ be the union of sets $\{e_{ij}, x_i, x_j, y_i, y_j\}$ over all $i, j$ violating P1, i.e., for which $e_{ij} = a$. We will show that we can recolor some vertices from $E'$ in order to obtain a homomorphism $h'$ with weight at least $p$, satisfying P1. We will use the iterative procedure described below; if the color of some vertex is not specified, it means that it is not changed.
Step 1. For any $e_{ij} \in E'$ such that $h(x_i) = a$ or $h(y_j) = a$ set $h'(e_{ij}) = b$ or $h'(e_{ij}) = c$, respectively. Observe that $h'$ is a homomorphism and clearly $w_{h'}(\{e_{ij}\}) > w_h(\{e_{ij}\})$, so the total weight is not decreased.

Step 2. For any $e_{ij} \in E'$ such that $h(e_{ij}) \in \{b, c\}$ observe that $h(x_i) = a$ or $h(y_j) = a$. If $h(x_i) = a$, then set $h'(y_j) = a$ and $h'(e_{ij}) = c$. Otherwise, set $h'(x_i) = a$ and $h'(e_{ij}) = b$. After that, go back to the Step 1. Note that $h'$ is a homomorphism and $w_{h'}(\{y_j, e_{ij}, x_i, e_{ij}\}) = w_h(\{y_j, e_{ij}, x_i, e_{ij}\})$.

Step 3. If there is still some $e_{ij} \in E'$, observe that $h(x_i) = h'(x_i) = b$ and $h(y_j) = h(y_j) = c$ and $h'(e_{ij}) = h'(x_i) = a$. Set $h'(x_i) = h'(y_j) = a$ and $h'(e_{ij}) = h'(e_{ij}) = b$, then go back to the Step 1. We note that $h'$ is a homomorphism and $w_{h'}(\{x_i, y_j, e_{ij}, e_{ij}\}) = w_h(\{x_i, y_j, e_{ij}, e_{ij}\})$.

We may change colors of vertices from $(X \cup Y) \cap E'$ more than once, but we always end $e_{ij}$ from $E'$ and thus the procedure terminates after at most $2m$ steps. Thus from now on we may assume that $P1$ holds for $h$.

Now assume that $P2$ does not hold, without loss of generality suppose that $h(x_i) = b$ and $h(y_j) = a$. Observe that $w_h(D_i) \leq 6$. Let $h'$ be obtained from $h$ by recoloring the segment $x_i$ to the color $a$, the segments $d_1, d_3, d_5, d_6$ to the color $b$, and the segments $d_2, d_4, d_7$ to the color $c$. Clearly, $h'$ is a homomorphism and its weight is at least the weight of $h$. Moreover, if $P1$ holds for $h$, then it holds in $h'$ as well. After at most $n$ such recolorings we obtain a homomorphism satisfying $P1$ and $P2$.

Finally, suppose that $P3$ is violated for some $i$. If $h(x_i) = h(y_j) = a$, we can safely color $d_2, d_4, d_5, d_6$ to $b$ and $d_3, d_7$ to $c$. If $h(x_i) = b$ and $h(y_j) = c$, then note that at least one segment from the gadget must be colored to $a$. We can color $d_7$ with $a$ and other segments from $D_i$ with $b$ and $c$. Note that this does not decrease the weight. We repeat this step while possible, and after at most $n$ steps we obtain a proper homomorphism.

Now let us show that $G$ has an independent set of size at least $k$ if and only if there exists a homomorphism $G^* \rightarrow H$ of weight at least $7n + 2m + k$. First, let $I$ be an independent set in $G$, such that $|I| \geq k$. Define a mapping $h : V(G^*) \rightarrow \{a, b, c\}$ as follows. For every $v_i \in I$, we color $d_7$ to $a$ and $x_i, d_3, d_4, d_6$ to $b$ and $y_i, d_1, d_2, d_5$ to $c$. For every $v_i \notin I$, we set $x_i, y_i$ to $a$ and $d_1, d_2, d_5, d_6$ to $b$ and $d_4, d_7$ to $c$. Note that for every edge $v_i v_j$ we either have $h(x_i) = a$ or $h(y_j) = b$, so we can always color $e_{ij}$ with the color $b$ or $c$. Observe that the weight of such defined $h$ is at least $2k + 6k + 7(n-k) + 2m = 7n + 2m + k$.

Now assume that there exists a homomorphism $h : G^* \rightarrow H$ of weight at least $7n + 2m + k$. By Claim 2 we can assume that $h$ is proper. Let $I$ be the set of vertices $v_i$ of $G$, such that $h(x_i) = b$ and $h(y_i) = c$. If $I$ is not independent, then there exist $v_i v_j \in E(G)$, such that $h(x_i) = h(x_j) = b$ and $h(y_i) = h(y_j) = c$. But then observe it implies that $h(e_{ij}) = h(e_{ji}) = a$, which contradicts the fact that $h$ is proper. Let $k'$ be the size of $I$ and compute the weight of $h$. Since $h$ is proper, every $e_{ij}$ contributes to the total weight. Moreover, $2k'$ vertices from $(X \cup Y)$ contribute to the weight. Finally, for every $v_i \in I$ we have $w_h(D_i) = 6$ and for every $v_i \notin I$ we have $w_h(D_i) = 7$. Summing up, the total weight of $h$ is $2m + 2k' + 6k + 7(n-k) = 7n + 2m + k'$. Since this is at least $7n + 2m + k$, we conclude that $k' \geq k$ and thus $I$ is an independent set with at least $k$ vertices.

Finally, we observe that $|V(G^*)| \leq 2n + 7n + 2m = O(n)$, which completes the proof.

2.2.3 Other cases

Now we show the hardness result, when the target graph is $C_4$, i.e., the graph $(c)$ in Figure 1.

Theorem 11. The WHOM($C_4$) problem in 2-DIR graphs with $n$ vertices cannot be solved in the time $2^{o(n)}$, unless the ETH fails.

Proof. Again, we will show a reduction from INDEPENDENT SET. The proof is analogous to the proof of Theorem 10 but simpler. However, this time we will make use of edge weights.

Let $a, b, c, d$ be consecutive vertices of $C_4$. We will build an instance $G^*$ of WHOM($C_4$). To define vertex weights, we set $w(v, a) = w(v, b) = 1$ and $w(v, c) = w(v, d) = 0$ for every $v \in V(G^*)$ (see Figure 8).

Let $(G, k)$ be an instance of INDEPENDENT SET with $V(G) = \{v_1, \ldots, v_n\}$ and $|E(G)| = m = O(n)$. Exactly like in section 2.2.2, we build a grid from two sets $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$, each containing $n$ disjoint parallel segments. There are no additional gadgets this time, only edge weights defined as follows. For every $i \in [n]$, we set $w(x_i y_i, e) = 0$ if $e \in \{ab, cd\}$ and $w(x_i y_i, e) = -\infty$ otherwise. For every $i, j$, such that $v_i v_j \in E(G)$, we set $w(x_i y_j, e) = 0$ if $e \in \{cb, cd, ad\}$ and $w(x_i y_j, ab) = -\infty$. All remaining edge weights are set to 0.
We claim that $G$ has an independent set of size at least $k$ if and only if there exists a weighted list homomorphism $h : G^* \to C_4$ of weight at least $2k$. Note that in order to obtain a homomorphism with positive weight, no edge can obtain the weight $-\infty$. So edge weights are actually edge lists. First, suppose there exists an independent set $I$ of size at least $k$ in $G$. We define a mapping $h$ as follows. For every $i$, such that $v_i \in I$, we set $h(x_i) = a$ and $h(y_i) = b$. For every $i$, such that $v_i \notin I$, we set $h(x_i) = c$ and $h(y_i) = d$. Note that $h$ is a homomorphism from $G^*$ to $C_4$. Moreover, if $v_i \in I$, then $h(x_i, y_i) = ab$ and if $v_i \notin I$, then $h(x_i, y_i) = cd$, so always $w_h(x_i, y_i) \neq -\infty$. Also, note that for every edge $x_i y_j$ of $G$ we have $h(x_i, y_j) \neq ab$, because $I$ is an independent set. This means that no edge gets weight $-\infty$ in $h$. Clearly the weight of $h$ is at least $2k$.

Now assume that there exists a homomorphism $h : G^* \to C_4$ with weight at least $2k$, respecting the lists. Observe that $G^*$ is bipartite, so one of its bipartition classes must be mapped to $\{a,c\}$ and another one to $\{b,d\}$. Without loss of generality assume that $h(X) \subseteq \{a,c\}$. Define a subset $I$ of the vertices of $G$ as follows: $v_i \in I$ if and only if $h(x_i) = a$. Let $k' := |I|$. We need to show that $I$ is an independent set and $k' \geq k$.

First, observe that for every $v_i \in V(G)$ we have that $h(x_i, y_i) \in \{ab, cd\}$, so $h(x_i) = a$ if and only if $h(y_i) = b$. This means there are $2k'$ vertices of $G^*$ mapped to $a$ or $b$. The weight of $h$ is at least $2k$, so $k' \geq k$. To show that $I$ is independent, assume that there exists $v_i, v_j \in E(G)$ such that $\{v_i, v_j\} \subseteq I$. It means that $b(x_i) = a$ and $b(y_j) = b$, but then the weight of $ab$ in $h$ is $-\infty$, a contradiction. So $G$ has an independent set of size at least $k$.

To complete the proof, note that graph $G^*$ has $2n$ vertices. \hfill $\square$

Note that Theorem 4 implies the hardness for all graphs $H$ that contain $C_4$ as a subgraph. It is because we can set the edge weights related to vertices and edges which do not belong to $C_4$ to $-\infty$. This completes the proof of Theorem 2(b).

3 Locally injective and locally bijective homomorphism

Now let us turn our attention to two other variants of the graph homomorphism problem, i.e., locally injective and locally bijective homomorphism. Recall that for a fixed $H$ (with possible loops on vertices), the LIHom($H$) (LBHom($H$), resp.) problem asks if a given graph $G$ admits a homomorphism $h$ to $H$ with a restriction that for every $v \in V(G)$, the mapping $h$ is injective (bijective, resp.) on the set $N_G(v)$. Local injectivity can be equivalently seen as “no two vertices of $G$ with a common neighbor may be mapped to the same vertex of $H$”. Moreover, every locally bijective homomorphism is also locally injective.

We show that unlike the WHOM($H$), both LIHom($H$) and LBHom($H$) can be solved in subexponential time on string graphs for every $H$.

**Theorem 4.** For every fixed graph $H$, the LIHom($H$) problem and the LBHom($H$) problem can be solved in time $2^{O(\sqrt{n} \log n)}$ in string graphs with $n$ vertices.

**Proof.** We will first prove the statement for LIHom($H$), and then for LBHom($H$).

**Locally injective homomorphisms.** To ensure consistency of solutions found in recursive calls, we will solve a more general problem. First, every vertex $v$ of $G$ is equipped with a list $L(v)$ of vertices of $H$, and we ask for a locally injective homomorphism respecting these lists. Second, we are given a subset $X$ of vertices of $G$ and a function $\sigma : X \to 2^{V(H)}$. We require that in a solution $h$, for every vertex $v \in X$ it holds that $h(N_G(v)) = \sigma(v)$, i.e., $\sigma(v)$ is the set of colors appearing on the neighbors of $v$. Clearly, if $L(v) = V(H)$ for every $v$, and $X = \emptyset$, then we obtain the LIHom($H$) problem. The algorithm will be recursive and the constraints related to the set $X$ will be checked at leaves of tree of recursive calls.

Observe that if $G$ has a vertex with degree larger than $|H|$, there is no way to map in an injective way. Thus in this case we immediately report a no-instance. So let us assume that every vertex has degree at most $|H|$, which means that $G$ has $O(|H|n)$ edges and thus, by Theorem 1, there is a balanced
separator $S$ of size $O(\sqrt{n}) = O(\sqrt{m})$, which can be found in time $2^{O(\sqrt{m} \log n)}$ by exhaustive search or in polynomial time, using the geometric representation. Let $V_1, V_2$ be sets such that $V(G) = V_1 \uplus V_2 \uplus S$, there is no $V_1$-$V_2$-path in $G - S$, and $V_1, V_2 \leq c' \cdot n$ for a constant $c'$.

Let $h$ be a fixed (unknown) solution. For each $v \in S$, we exhaustively guess the color $h(v)$ (respecting the list $L(v)$) and the sets $\sigma_1 := h(N_G(v) \cap V_1)$ and $\sigma_2 := h(N_G(v) \cap V_2)$, respecting $\sigma(v)$, if $v \in X$. Note that, as $h$ is locally injective, $\sigma_1$ and $\sigma_2$ are disjoint. Then recursively solve the problem in $G_1 := G[V_1 \cup S]$ and $G_2 := G[V_2 \cup S]$. In the recursive calls we set $L(v) = \{h(v)\}$ for every $v \in S$. Moreover, we include each such $v$ in $X$ and set $\sigma(v) = \sigma_1$ in the recursive call for $G_1$ and $\sigma(v) = \sigma_2$ in the call for $G_2$. This ensures that no vertex from $S$ has two neighbors of the same color, one in each of $G_1, G_2$. The total number of recursive calls is at most $(|H| \cdot 3^{|H|}|S| = 2^{O(\sqrt{m})}$ and the complexity of the whole algorithm is also $2^{O(\sqrt{m} \log n)}$.

**Locally bijective homomorphisms.** The proof follows very similarly to the previous case. It varies only in the part in which we guess the colors of vertices belonging to the separator $S$. Again, let $h$ be a fixed, unknown solution. Observe that as $h$ is locally bijective, each vertex $v$ of $G$ must be mapped by $h$ to a vertex of equal degree. To ensure that, when guessing the color of $v \in S$, we just remove from $L(v)$ all elements $a$ for which $\deg_G(v) \neq \deg_H(a)$.

As mentioned before, locally injective homomorphisms generalize some well-studied graph labeling problems, so **Theorem 4** implies the following.

**Corollary 12.** For any fixed $k$, (i) the $k$-$L(2,1)$-labeling, (ii) the $k$-coloring of the square of a graph, (iii) the injective $k$-coloring, can be solved in time $2^{O(\sqrt{m})}$ in string graphs.

On the other hand, since every planar graph is a segment graph [3][13], hardness results for planar graphs can be used to derive ETH-lower bounds for LIHom$(H)$ in particular, the following lower bounds follow from the hardness results for $k$-$L(2,1)$-labeling [9], 7-coloring of the square of a graph [36], and injective 3-coloring [2].

**Theorem 13** (Eggemann, Havet, Noble, [9], Ramanathan, Lloyd [36], Bertossi, Bonuccelli [2]). Unless the ETH fails, there is no algorithm for LIHom$(H)$ in segment graphs working in time $2^{o(\sqrt{m})}$, where $H$ is (i) the complement of a path with at least 4 vertices, (ii) a complete graph with 7 vertices, (iii) a triangle with additional loop on its every vertex.

## 4 Locally surjective homomorphism

In this section we consider the problem of locally surjective homomorphism, denoted by LSHom$(H)$. For a fixed graph $H$, the LSHom$(H)$ problem asks whether a given graph $G$ admits homomorphism to $H$, which is surjective on $N_G(v)$ for every $v \in V(G)$. In other words, if $h(v) = a$, then every neighbor of $a$ appears on a neighbor of $v$. If there exists a locally surjective homomorphism from $G$ to $H$, we denote it by $G \rightarrow_{LS} H$. For a homomorphism $h : G \rightarrow H$ we say that a vertex $v$ is happy if $h(N_G(v)) = N_H(h(v))$. Clearly $h$ is locally surjective if every vertex of $G$ is happy.

We aim to prove **Theorem 5** i.e., prove a dichotomy for simple target graphs $H$ with $\Delta(H) \leq 2$. Let us start with a simple observation, that will be used many times.

**Observation 14.** Let $h : G \rightarrow_{LS} H$ and $\delta(H) \geq 1$. For every $v \in V(G)$ it holds that:

a) $\deg_G(v) \geq \deg_H(h(v))$. In particular, if $\deg_G(v) = 1$, then $\deg_H(h(v)) = 1$,

b) if $\deg_G(v) = \deg_H(h(v)) = 2$, then $h(v_1) \neq h(v_2)$ for distinct neighbors $v_1, v_2$ of $v$.

### 4.1 Paths

In this section, the the target graph is a path $P_k$ with consecutive vertices 1, 2, ..., $k$.

First, let us discuss the case, when $k = 3$. Let $G$ be an instance of LSHom$(P_3)$. By **Observation 14** we can assume that $G$ does not have isolated vertices. We can also assume that an input graph $G$ is bipartite with bipartition classes $X$ and $Y$, as otherwise any homomorphism to $P_3$ cannot exist. Moreover, in any homomorphism, one of bipartition classes, say $Y$, will be entirely mapped to 2. Note that since no vertex is isolated, vertices of $X$ will always be happy. Thus $G \rightarrow_{LS} P_3$ if and only if one can color vertices of $X$ with two colors (1 and 3), so that every vertex from $Y$ has at least one neighbor in each color. We observe that this is exactly the NOT ALL EQUAL SAT problem, where $G$ is an incidence graph of the input formula. From this we conclude that LSHom$(P_3)$ does not have a subexponential
algorithm in general graphs, but is solvable in polynomial time in planar graphs (since \textsc{Planar Not All Equal Sat} is in \textsc{P}, see Moret \cite{moret2000}). Moreover, the list variant of $\text{LSHom}(P_3)$ in planar graphs is \textsc{NP}-complete, see Dehghan \cite{dehghan2005}.

The win-win approach of \textbf{Theorem 2} (a) cannot be directly applied for $\text{LSHom}(P_3)$, as there is no good branching on a high-degree vertex. Instead, we will use the following result.

\textbf{Theorem 15 (Lee \cite{lee2007}).} There is a constant $c > 0$ such that for every $t \geq 1$, it holds that every $K_{1,t}$-free string graph on $n$ vertices has at most $c \cdot n \cdot t \log t$ edges.

Now we can show that $\text{LSHom}(P_3)$ can be solved in subexponential time in string graphs.

\textbf{Theorem 16.} $\text{LSHom}(P_3)$ can be solved in time $2^{O(n^{2/3} \log^{2/3} n)}$ for string graph on $n$ vertices, even if geometric representation is not given.

\textbf{Proof.} We assume an instance graph $G$ has no isolated vertices and is bipartite, with bipartition classes $X$ and $Y$. Note that $G$ is a yes-instance if and only if there is a homomorphism $h_X : G \to P_3$, such that $h_X(X) = \{1,3\}$ and $h_X(Y) = \{2\}$, or homomorphism $h_Y : G \to P_3$, such that $h_Y(Y) = \{1,3\}$ and $h_Y(X) = \{2\}$. Let us assume that $X$ is mapped to $\{1,3\}$, the algorithm will be called twice with roles of $X$ and $Y$ switched. Again, we will solve a more general problem, in which we define an additional function $\sigma : Y \to 2^{\{1,3\}}$ and ask for an existence of a homomorphism $h : G \to P_3$, such that $\sigma(y) \subseteq h(N_G(y))$ for every $y \in Y$. Clearly, if $\sigma \equiv \{1,3\}$, then we obtain the $\text{LSHom}(P_3)$ problem.

First, we observe that if $G$ has at most $\frac{2}{3} \cdot n^{4/3} \log n$ edges (where $c$ is a constant from \textbf{Theorem 15}), then we can find a balanced separator $S$ of size $O(n^{2/3} \log^{1/2} n)$ in time $2^{O(n^{2/3} \log^{2/3} n)}$. Denote by $V_1, V_2$ the sets such that $V(G) = V_1 \uplus V_2 \uplus S$ and there is no $V_1$-$V_2$-path in $G - S$. We exhaustively guess $h(x)$ for every $x \in S \cap X$ and the partition $\sigma_1 \uplus \sigma_2$ of $\sigma(y)$, $h(N_G(y) \cap S)$ for every $y \in S \cap Y$. Then, for every possibility, we consider graphs $G_1 := G[V_1 \uplus S]$ and $G_2 := [V_2 \uplus S]$, in which vertices of $S$ are already colored. For every $y \in Y \cap S$ we set $\sigma(y) = \sigma_1$ if $y \in V(G_1)$ or $\sigma(y) = \sigma_2$ otherwise. Then, for every $x \in X \cap S$, we remove $h(x)$ from $\sigma(y)$, for every neighbor $y$ of $x$, and finally we remove $x$ from the instance. If there exists $y$ for which $\sigma(y) = \emptyset$, we also remove $y$. Then, if any isolated vertex $x \in X$ appears, we remove it too, as it means that $\sigma(y)$ of its every neighbor $y$ was already empty, so the color of $x$ does not matter. We call the algorithm recursively for graphs $G_1$ and $G_2$, together with their corresponding functions $\sigma$. Note that $G_1$ or $G_2$ may contains an isolated vertex $y \in Y$ with $\sigma(y) \neq \emptyset$, in this case we terminate the current recursive call. Observe that the total number of recursive calls is $2^{|X \cap S|} \cdot 4^{|Y \cap S|} = 2^{O(n^{2/3} \log^{1/3} n)}$ and the overall complexity of this step is $2^{O(n^{2/3} \log^{2/3} n)}$.

If $G$ has more than $\frac{2}{3} \cdot n^{4/3} \log n$ edges, we know from \textbf{Theorem 15} that it has a bipartite subgraph $K_{n^{1/3}, n^{1/3}}$. We find it exhaustively in time $n^{O(n^{1/3})} = 2^{O(n^{2/3} \log n)}$. Let $X' \subseteq X$ and $Y' \subseteq Y$ be its bipartition classes. We branch on three possibilities. Either we set $h(x) = 1$ for every $x \in X'$, or $h(x) = 3$ for every $x \in X'$ or we choose $x_1, x_2 \in X'$ and set $h(x_1) = 1$ and $h(x_2) = 3$. In first two cases we can proceed to the graph $G - X'$ (and remove $h(X')$ from $\sigma(y)$ of every $y \in N(X')$), and in the last case we can remove $Y'$, together with $x_1$ and $x_2$ (also adjusting the function $\sigma$ for their neighbors), as all elements of $Y'$ are happy. Denote by $F(n)$ the complexity of this step and observe that

$$F(n) \leq 2^{O(n^{2/3} \log n)} + 2F(n - n^{1/3}) + n^{2/3}F(n - n^{1/3}) \leq 2^{O(n^{2/3} \log n)},$$

so the total complexity of algorithm is also $2^{O(n^{2/3} \log^{2/3} n)}$, \qed

For paths with at least 4 vertices, the existence of subexponential algorithms are unlikely.

\textbf{Theorem 17.} For any $k \geq 4$, the $\text{LSHom}(P_k)$ problem on 2-DIR graphs with $n$ vertices cannot be solved in time $2^{o(n)}$, unless the ETH fails.

\textbf{Proof.} Let $k \geq 4$ be fixed. We reduce from 3-SAT, consider an instance $\Phi$ of 3-SAT with variables $u_1, \ldots, u_n$ and clauses $c_1, \ldots, c_m$. We assume that every variable $u$ appears at least once as positive and once as negative literal. Indeed, otherwise we can set its value and remove it from the formula, along with all clauses containing $u$.

For each occurrence of a variable $u_i$ in a clause, we introduce a vertical occurrence segment $x$. We denote the sets of positive and negative occurrence segments of $u_i$ by $X_i$ and $\bar{X}_i$ respectively. Let $X = \bigcup_{i \in [k]} (X_i \uplus \bar{X}_i)$. We place the elements of $X$ in a following order: leftmost segments from $X_1$, then the ones from $\bar{X}_1, X_2, \bar{X}_2$, etc. Moreover, each segment is slightly shorter than the one on its left (see Figure \ref{fig:12} (a)). For each clause $c_p$, we introduce a horizontal clause segment $y_p$, intersecting all
Theorem 18. Let \( Y = \bigcup_{p \in [n]} y_p \). We also add a vertical path \( T = (t_1, \ldots, t_{2k-1}) \) on the right side of the picture. For each \( y \in Y \) we introduce a horizontal segment \( y' \) on the right side of \( y \), which intersects \( y \) and \( t_1 \). Let \( Y' \) be the set of all these segments \( y' \). For every \( x \in X \) we add a horizontal path \((q_1, \ldots, q_{k-3})\) such that \( q_1 \) intersects \( x \) and \( q_{k-3} \) intersects \( t_k \).

For each \( u_i \) we add a **variable gadget** on top of its occurrence segments. It is a path \((r_1^i, \ldots, r_{2k-3}^i)\) such that \( r_1^i \) (or \( r_{2k-3}^i \), resp.) intersects all segments from \( X_i \) (\( X_i \), resp., see Figure 9 (a)). Now consider an occurrence segment \( x_i \), and let \( c_p \) be the clause containing this particular occurrence of a variable. On the intersection of \( x_i \) with \( y_p \) we introduce a **membership gadget**, containing two segments \( \alpha_i \) and \( \beta_i \), crossing each other and \( x_i \) or \( y_p \), resp. Also, we add a path \((s_1, \ldots, s_{2k-4})\) such that \( s_1 \) and \( s_{2k-4} \) also intersect \( \alpha_i \) and \( \beta_i \), resp. (see Figure 9 (b)). Let \( E' \) be the set of segments of the membership gadget of \( x_i \).

Assume that there exists \( h : G \rightarrow P_k \). We will show the satisfying truth assignment \( \varphi \) of \( \Phi \). First, note that without loss of generality \( h(t_{2k-1}) = 1 \), as \( \deg(t_{2k-1}) = 1 \). From Obs. (14) we get that \( h(t_k) = k \) and \( h(t_1) = 1 \). It implies that \( h(Y') = 2 \) and, by Obs. (14) b, \( h(Y) = 3 \). Moreover, as \( h(t_k) = k \), then for every \( x \in X \) its corresponding path between \( x \) and \( t_k \) can be colored in only one way, and \( h(X) = 2 \).

For every \( v_i \in V(G) \) it holds that \( \{h(r_1^i), h(r_{2k-3}^i)\} = \{1, 3\} \). Define \( \varphi \) as follows: if \( h(r_1^i) = 1 \), then \( \varphi(u_i) = 1 \), if \( h(r_{2k-3}^i) = 1 \), then \( \varphi(u_i) = 0 \). Assume that there exists a clause \( c_p \), which is not satisfied by \( \varphi \), and let \( x_i, x_j, x_l \) be the occurrence segments corresponding to the literals of \( c_p \). As all literals of \( c_p \) are false, the neighbors of \( x_i, x_j, x_l \) in their variable gadgets are mapped to 3. To make \( x_i, x_j, x_l \) happy, we need to have \( h(\alpha_i) = h(\alpha_j) = h(\alpha_l) = 1 \), which implies that \( h(\beta_i) = h(\beta_j) = h(\beta_l) = 2 \). But then \( y_p \) is not happy, a contradiction.

Now we show that if \( \Phi \) has a satisfying assignment \( \varphi \), then there exists \( h : G \rightarrow P_k \). We set the coloring \( h \) of all segments of \( G \), which do not belong to variable or membership gadgets, exactly like in previous step. If \( \varphi(u_i) = 1 \), we set \( h(r_1^i) = 1 \) and \( h(r_{2k-3}^i) = 3 \). Otherwise, \( h(r_1^i) = 3 \) and \( h(r_{2k-3}^i) = 1 \). In both there is only way to color the path \((r_1^i, \ldots, r_{2k-4}^i)\) in a locally surjective way.

Observe that every \( x_i \in X \) which is adjacent to \( r_1^i \) if \( \varphi(u_i) = 1 \) or to \( r_{2k-3}^i \) if \( \varphi(u_i) = 0 \) is already happy. In this case we set \( h(\alpha_i) = 3 \) and \( h(\beta_i) = 4 \), and color \( s_1, s_2, \ldots, s_{2k-4} \) with \( 2, 1, 2, \ldots, k-1, k, k-1, \ldots, 5 \) (or \( 2, 1, 2, 3 \) for \( k = 4 \)). If the neighbor of \( x_i \in X \) is not mapped to 1 (i.e., \( x_i \) corresponds to a false literal), we must make it happy by setting \( h(\alpha_i) = 1 \), which implies that \( h(\beta_i) = 2 \). We color the path \( s_1, \ldots, s_{2k-4} \) with \( 2, 3, \ldots, k-1, k, k-1, \ldots, 3 \). This makes all segments inside \( E' \) happy.

Finally, observe that each since \( \varphi \) is a satisfying assignment, each clause \( c_p \) contains a true literal \( x_i \), and thus \( y_p \) has a neighbor \( \beta_i \) mapped to 4, so each variable segment is happy. This means that \( h \) is a locally surjective homomorphism. □

### 4.2 Cycles

In this section we assume that the target graph is a cycle with consecutive vertices \( 1, 2, \ldots, k \) (so 1 is adjacent to \( k \)). Let us start with the case \( k = 4 \).

Theorem 18. \( \text{LSHom}(C_4) \) can be solved in time \( 2^{O(n^{2/3} \log^{2/3} n)} \) for string graph on \( n \) vertices, even if a geometric representation is not given.

**Proof.** Again, we assume that an instance graph \( G \) is bipartite with bipartition classes \( X \) and \( Y \), without isolated vertices. Clearly in any solution \( h \) we either have \( h(X) = \{1, 3\} \) and \( h(Y) = \{2, 4\} \), or \( h(X) = \{2, 4\} \) and \( h(Y) = \{1, 3\} \).
Let us show that there exists $h : G \xrightarrow{\phi} C_4$ such that $h(X) = \{1, 3\}$ if and only if there exist $h_1, h_2 : G \xrightarrow{\phi} P_3$ for which $h_1(X) = h_2(Y) = \{1, 3\}$ and $h_1(Y) = h_2(X) = \{2\}$. First, consider $h : G \xrightarrow{\phi} C_4$ such that $h(X) = \{1, 3\}$, which implies that $h(Y) = \{2, 4\}$. Define $h_1(z) = 2$ if $z \in Y$ and $h_1(z) = h(z)$ otherwise. Define $h_2(z) = 2$ if $z \in X$ and $h_1(z) = h(z) - 1$ otherwise. Clearly, $h_1, h_2 : G \rightarrow P_3$ and $h_1(X) = h_2(Y) = \{1, 3\}$. Assume $h_1$ is not locally surjective, i.e., there exists $v$ which is not happy. Note that if $v \in X$, then $v$ must be an isolated vertex, a contradiction. If $v \in Y$ and, without loss of generality, $h_1(N(v)) = \{1\}$, this means that $h(N(v)) = \{1\}$, again, a contradiction, because $h$ is locally surjective. Analogous argument works for $h_2$.

Now assume that there exist $h_1, h_2 : G \xrightarrow{\phi} P_3$ such that $h_1(X) = h_2(Y) = \{1, 3\}$. For every $z \in X$ let $h(z) := h_1(z)$ and for every $z \in Y$ let $h(z) := h_2(z) + 1$. Observe that $h(X) = \{1, 3\}$ and $h(Y) = \{2, 4\}$, so $h$ is a homomorphism. Assume it is not locally surjective, and $v$ is not happy. Without loss of generality let $h(v) = 2$ and $h(N(v)) = \{1\}$. This means that $h_1(v) = 2$ and, as $h_1$ was locally surjective, there is $u \in N(v)$ for which $h_1(u) = 3$. But $u \in X$, so $h_1(u) = h(u) = 3$, a contradiction.

To solve $\text{LSHom}(C_4)$, we run the algorithm from Theorem 16 twice, switching the roles of $X$ and $Y$. We return true for $\text{LSHom}(C_4)$ if both calls return true.

The total running time is $2^{O(n^{2/3} \log^{3/2} n)}$.

It appears that existence of subexponential algorithms for remaining $k$ is unlikely.

**Theorem 19.** Let $k \geq 3$, $k \neq 4$. There is no algorithm solving $\text{LSHom}(C_k)$ on a 2-DIR graph with $n$ vertices in time $2^{o(n)}$, unless the ETH fails.

**Proof.** Again, we reduce from 3-SAT. Assume that every variable $u$ appears at least once as a positive and once as a negative literal. Our construction is similar to the one in the proof of Theorem 17, we just change the length of some paths. We construct the sets $X$, $Y$ and $Y'$ in the same way. Each variable gadget is now a path $(r_1^1, \ldots, r_{k-1}^1)$ such that $r_1^1$ and $r_{k-1}^1$ intersect all segments from $X_i$ and $X_{i'}$ respectively. In each membership gadget $E'$ the path between $\alpha_i$ and $\beta_i$ has now $k - 2$ segments. Also the path $T$ has length $k - 2$ and still each $y' \in Y'$ intersects $t_1$. For every $x \in X$, we add a single segment $x'$, intersecting $x$ and $t_{i-2}$ (instead of a path, see Figure 9(a)). Denote the set of these segments $x'$ by $X'$.

We show that $G \xrightarrow{\Phi} C_k$ iff $\Phi$ is satisfiable. Assume that there exists $h : G \xrightarrow{\Phi} C_k$. Let us start with analyzing its structure, the argument will be split into two cases.

First consider $k \geq 5$. By symmetry of $C_k$, we can assume that $h(t_1) = 1$ and $h(t_2) = k$. By Observation 14(b), it implies that $h(t_{k-2}) = 4$. Vertices $t_1$ and $t_{k-2}$ must be happy, so there exist $x' \in X'$ and $y' \in Y'$, such that $h(x') = 3$ and $h(y') = 2$, which means $h(x) = 2$ and $h(y) = 3$ for their corresponding neighbors $x \in X$ and $y \in Y$. Note that if there exists $x' \in X'$ such that $h(z') = 5$, then its neighbor $z$ from $X$ must be mapped to 6 (or 1, if $k = 5$). However, then $h(x)$ is not a neighbor of $h(y) = 3$, a contradiction.

If $k = 3$, then, by symmetry, we assume that $h(t_1) = 1$ and $h(y') = 2$ for some $y' \in Y'$. By Obs. 14(b), note that $h(y) = 3$ for the neighbor $y$ of $y'$. If there is some $x' \in X'$, such that $h(x') = 2$, then its neighbor $x$ must be mapped to 3, which is impossible since $h(y) = 3$. 

![Figure 10: An overall construction of $G$ for (a) path $P_5$ and (b) cycle $C_5$. The red segments stand for $T$, blue ones are the elements of $Y'$. Black rectangles are variable gadgets with details shown on Figure 9(a). Violet squares are membership gadgets with details shown on Figure 9(b).](image-url)
In both cases we obtain that every segment from $X'$ is mapped to 3 and every segment from $X$ is mapped to 2. Analogously we can show that $h(Y') = 2$ and $h(Y) = 3$.

For each $i$, we have $\{h(r_i^1), h(r_i^{1}_{k-1})\} = \{1, 3\}$. We define $\varphi(u_i) = 1$ if $h(r_i^1) = 1$, otherwise $\varphi(u_i) = 0$. Suppose that $\varphi$ does not satisfy $\Phi$. Let $e_p$ be an unsatisfied clause. Since $h$ is locally surjective, $y_p$ is happy, so it has a neighbor $\beta_i$ such that $h(\beta_i) = 4$ (or $h(\beta_i) = 1$ if $k = 3$). It implies that $h(\alpha_i) = 3$ and thus the neighbor of the occurrence segment $x_i$ in the variable gadget is mapped to 1. Therefore, $x_i$ corresponds to a true literal, a contradiction.

Now assume that $\varphi$ is a satisfying assignment for $\Phi$. We define the coloring $h$ of all vertices of $G$ except the ones in variable or membership gadgets exactly as above. For each variable $u_i$, if $\varphi(u_i) = 1$, we set $h(r_i^1) = 1$, otherwise $h(r_i^{1}_{k-1}) = 1$. We color remaining vertices of vertex gadgets in the only possible way. Observe that every $x_i \in X$ which has a neighbor $r_i^1$ or $r_i^{1}_{k-1}$ mapped to 1 is already happy, so we can set $h(\alpha_i) = 3$ and $h(\beta_i) = 4$ (or $h(\beta_i) = 1$ if $k = 3$), and color all remaining vertices of $H'$ such that $h(E') = |k|$. Such vertices $x_i$ corresponds to true literals. If $x_i$ still has no neighbors mapped to 1, we need to set $h(\alpha_i) = 1$, which implies $h(\beta_i) = 2$. Note that this coloring can be extended to the remaining segments in the membership gadget. Observe that a clause segment $y_p$ is happy only if it has a neighbor $\beta_i$ mapped to 4 (or 1 for $k = 3$), and recall that for such $\beta_i$, the segment $x_i$ corresponds to a true literal. As $\varphi$ is a satisfying assignment, such literal exists in each clause, so all vertices of $Y$ must be happy, which means $h$ is locally surjective.

4.3 One more hard graph

Finally, let us consider the graph $H$ in Fig. 11 (left). We will show the following.

![Figure 11: A graph $H$ (left) and a clause gadget (right).](image)

**Theorem 20.** Assuming the ETH, there is no algorithm solving the LSHOM($H$) on a segment graph with $n$ vertices in time $2^{o(n)}$.

**Proof.** We reduce from 3-SAT. Consider a 3-SAT formula $\Phi$ with variables $u_1, u_2, \ldots, u_n$ and clauses $c_1, c_2, \ldots, c_m$, each of which is an alternative of exactly three literals. Again, we may assume that each variable $u_i$ appears at least once as a positive and at least once as a negative literal.

Let us construct a segment graph $G$, which is an instance of LSHOM($H$). For each variable $u_i$, we introduce two variable segments $x_i$ and $y_i$, intersecting each other. The segment $x_i$ will correspond to positive appearances of $u_i$, while $y_i$ will correspond to the negative ones. For each clause we introduce a clause gadget depicted in Fig. 11 (right). The segments $p_1, p_2, p_3$ correspond to literals of the clause.

For every appearance of $u_i$ in a clause $c_j$, we add an occurrence segment intersecting the appropriate segment of $x_i, y_i$ and one of $p_1, p_2, p_3$ in the gadget encoding the clause $c_j$. The occurrence segments do not intersect other variable segments and segments in clause gadgets, but may intersect each other. Finally, for every occurrence segment $s$ we add two segments $e$ and $f$, such that $f$ intersects only $e$, and $e$ intersects only $s$ and $f$. The overall picture of the construction is shown in Fig. 12.

Let us show that $\Phi$ is satisfiable if and only if $G$ has a locally surjective homomorphism to $H$. First, suppose that $\Phi$ is satisfiable and let $\varphi$ be a satisfying truth assignment. Let us define a mapping $h: V(G) \rightarrow \{a, b\}$. For each variable $u_i$, if $\varphi(u_i)$ is true, then $h(x_i) = a$ and $h(y_i) = b$, otherwise $h(x_i) = b$ and $h(y_i) = a$. For each occurrence segment $s$, and its corresponding segments $e, f$, we set $h(s) = b, h(e) = b$, and $h(f) = a$. Now consider a clause gadget corresponding to a clause $c_j$. Since $\varphi$ is a satisfying assignment, $c_j$ has at least one true literal, let is be $k$-th literal in $c_j$ for $k \in \{1, 2, 3\}$. We set $h(p_k) = b$ and $h(q_k) = a$. Moreover, we set $h(p_i) = a$ and $h(q_i) = b$ for $i \neq k$. Finally, we set $h(z) = b$. It is straightforward to see that $h$ is a homomorphism to $H$. Let us now show that it is locally surjective.

First, we observe that each variable segment is happy. Indeed, recall that $h(x_i) \neq h(y_i)$ and that each of $x_i, y_i$ intersects at least one occurrence segment, which is mapped to $b$. Now consider an occurrence segment $s$ with its corresponding segments $e, f$. The segment $f$ is happy, because it is adjacent to $e$, which is mapped to $b$. The segment $e$ is also happy, since it is adjacent to $f$ and $s$, which are mapped
to $a$ and $b$, respectively. Moreover, $s$ is adjacent to $e$, so to make it happy, it needs to be adjacent to a vertex mapped to $a$. If the literal corresponding to $s$ is true, then such a vertex is either $x_i$ or $y_i$. So assume that $s$ corresponds to a literal that is false. Note that this literal is not satisfying any clause, so $s$ intersects some $p_k$ in a clause gadget, such that $h(p_k) = a$. Therefore $s$ is always happy. Finally, since each occurrence segment is mapped to $b$, it is straightforward to see that each segment in a clause gadget is also happy. This shows that $h$ is locally surjective.

For the other direction, suppose that $h$ is a locally surjective homomorphism from $G$ to $H$. Consider an occurrence segment $s$ with its corresponding segments $e, f$. Note that since $f$ is happy, it must that $h(f) = a$ and $h(e) = b$. Now, since $e$ is happy, we must have $h(s) = b$. Now consider the variable segments for a variable $u_i$. Note that they only intersect each other and occurrence segments. Thus, to make them happy, one of $x_i, y_i$ must be mapped to $a$ and the other one to $b$. For each variable $u_i$, we set $\varphi(u_i)$ true if an only if $h(x_i) = a$. Let us show that $\varphi$ satisfies $\Phi$. Suppose the contrary, i.e., there is a clause $c_j$ which is not satisfied by $\varphi$, i.e., all segments corresponding to literals of $c_j$ are mapped to $b$. Consider the segments in the clause gadget corresponding to $c_j$. Note that in order to make the occurrence segments happy, we need to set $h(p_1) = h(p_2) = h(p_3) = a$. Since $h$ is a homomorphism, we need to have $h(q_1) = h(q_2) = h(q_3) = b$. Now, to make $q$’s happy, we need to have $h(z) = b$. However, this way $z$ is not adjacent to any segment mapped to $a$, so it is unhappy, a contradiction.

5 Concluding remarks and open problems

5.1 $P_7$-free graphs

Let us start with proving Theorem 3.

Theorem 3. Let $H$ be a fixed graph.

(a) If $H$ has no two vertices with a common neighbor, then for all fixed $t$ the WHOM($H$) problem can be solved in time $2^{O(\sqrt{n})}$ for $P_t$-free graphs with $n$ vertices.

(b) Otherwise, there is no algorithm solving the WHOM($H$) problem in time $2^{o(n)}$ for $P_t$-free graphs with $n$ vertices, unless the ETH fails.

Proof. Recall that part (a) was proven by Groenland et al. [20]. We will show that the proof of Theorem 2 (b) implies Theorem 3 (b). Let us consider again the problem WHOM($H$), for $H$ shown in Figure 1(a). We go back to the proof of Theorem 9 and observe that the longest induced path of each instance $G^*$ has at most 6 vertices (if $v_i, v_i', v_j, v_j'$ are disjoint edges of $G$, then it is the path $\beta_{ii'}, \alpha_{ii'}, x_i, y_j, \alpha_{jj'}, \beta_{jj'}$). Clearly, this means that there is no algorithm solving MAX CUT (and thus WHOM($H$) for $H$ in Figure 1(a)) in time $2^{o(n)}$ on $P_7$-free graphs on $n$ vertices, unless the ETH fails. Moreover, if instead of gadgets we used edge-weights, as in the proof of Theorem 11 we obtain hardness of WHOM($H$) for complete
graphs. Note that complete graphs are $P_3$-free, and clearly the problem is polynomially solvable on $P_3$-free graphs.

Analogously we can conclude that, assuming the ETH, there is no subexponential algorithm for WHom$(H)$ for $H$ shown in Figure 1(b). Note that the instance constructed in the proof of Theorem 10 is always $P_3$-free, and substituting gadgets with appropriate edge weight gives the hardness for complete bipartite graphs. For the remaining graphs in Figure 1, the instance constructed in the proof of Theorem 11 is also complete bipartite. Note that complete bipartite graphs are $P_3$-free, and for all graphs $H$ in Figures 1(b) – (g), WHom$(H)$ is polynomially solvable for $P_3$-free graphs.

Let us also point out that applying the approach of Theorem 11 to a $P_1$-free graph yields a polynomial algorithm (for fixed $H$). Indeed, a $P_1$-free graph with maximum degree at most $|H|$ has at most $t \cdot |H|$ vertices, which is a constant, and thus the problem can be brute-forced in constant time.

Moreover, we observe that also proofs in the Section 5 give corollaries for $P_t$-free graphs. Indeed, the graphs constructed in Theorem 5 are $P_t$ free for some $t$ (depending on $k$). The longest induced path in the graph constructed in the proof of Theorem 17 has at most $10k$ vertices: it contains four vertices from $X$, $2(k - 1)$ vertices from two variable gadgets, $2(k - 3)$ vertices from two paths joining elements of $X$ with $t_k$, $2(2k - 3)$ vertices from two membership gadgets and the vertex $t_k$ itself. The longest induced path in graph constructed in the proof of Theorem 19 has $4k + 3$ vertices: again, four vertices from $X$, $2(k - 1)$ vertices from two variable gadgets, two vertices from $X'$, $2(k - 1)$ vertices from two membership gadgets and the vertex $t_k$ itself. From this we conclude that if $H$ is an irreflexive graph with $\Delta(G) \leq 2$ then for some constant $t$ the subexponential algorithm for LSHom$(H)$ for $P_t$-free graphs does not exist, unless the ETH fails. Finally, note that the construction in the proof of Theorem 20 can be modified so that all vertices corresponding to occurrence segments form a clique. After this modification the graph might not be a segment graph anymore, but it is $P_{12}$-free.

5.2 Further research directions

Let us conclude the paper with pointing out some directions for further research. First, it would be interesting to obtain a complexity dichotomy for the problems of finding a homomorphism and a list homomorphism from a string graph to a fixed graph $H$.

We believe that the problem can be solved in subexponential time for some graphs $H$ that do not have property (⋆).

Next, we think that obtaining a full complexity dichotomy for LSHom$(H)$ in string graphs is an exciting (and probably difficult) task. Let us mention that the NP-hardness proof by Fiala and Paulusma [13] implies that if $H$ is a connected graph with at least two edges, then LSHom$(H)$ cannot be solved in subexponential time in general graphs.

Finally, recall that our hardness proofs for LSHom$(H)$ imply hardness for LSHom$(H)$ in $P_t$-free graphs. We think it is interesting whether LSHom$(P_3)$ (and thus LSHom$(C_4)$, as they are closely related) can be solved in subexponential time in $P_t$-free graphs.

References

[1] J. Alber and J. Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. J. Algorithms, 52(2):134–151, 2004.
[2] A. A. Bertossi and M. A. Bonuccelli. Code assignment for hidden terminal interference avoidance in multihop packet radio networks. IEEE/ACM Transactions on Networking, 3(4):441–449, Aug 1995.
[3] C. Biró, É. Bonnet, D. Marx, T. Miltzow, and P. Rzążewski. Fine-grained complexity of coloring unit disks and balls. In B. Aronov and M. J. Katz, editors, 33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia, volume 77 of LIPIcs, pages 18:1–18:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
[4] É. Bonnet and P. Rzążewski. Optimality program in segment and string graphs. CoRR, abs/1712.08907, 2017.
[5] É. Bonnet and P. Rzążewski. Optimality program in segment and string graphs. In A. Brandstädt, E. Köhler, and K. Meer, editors, Graph-Theoretic Concepts in Computer Science - 44th International Workshop, WG 2018, Cottbus, Germany, June 27-29, 2018, Proceedings, volume 11159 of Lecture Notes in Computer Science, pages 79–90. Springer, 2018.
[6] J. Chalopin and D. Gonçalves. Every planar graph is the intersection graph of segments in the plane: extended abstract. In Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009, pages 631–638, 2009.

[7] S. Chaplick, J. Fiala, P. van ‘t Hof, D. Paulusma, and M. Tesar. Locally constrained homomorphisms on graphs of bounded treewidth and bounded degree. Theor. Comput. Sci., 590:86–95, 2015.

[8] A. Dehghan. On strongly planar not-all-equal 3SAT. Journal of Combinatorial Optimization, 32(3):721–724, Oct 2016.

[9] N. Eggemann, F. Havet, and S. D. Noble. k-L(2,1)-labelling for planar graphs is NP-complete for k ≥ 4. Discrete Applied Mathematics, 158(16):1777–1788, 2010.

[10] T. Feder, P. Hell, and J. Huang. List homomorphisms and circular arc graphs. Combinatorica, 19(4):487–505, 1999.

[11] J. Fiala and J. Kratochvíl. Locally injective graph homomorphism: Lists guarantee dichotomy. In F. V. Fomin, editor, Graph-Theoretic Concepts in Computer Science, 32nd International Workshop, WG 2006, Bergen, Norway, June 22-24, 2006, Revised Papers, volume 4271 of Lecture Notes in Computer Science, pages 15–26. Springer, 2006.

[12] J. Fiala and J. Maxová. Cantor-Bernstein type theorem for locally constrained graph homomorphisms. Eur. J. Comb., 27(7):1111–1116, 2006.

[13] J. Fiala and D. Paulusma. A complete complexity classification of the role assignment problem. Theor. Comput. Sci., 349(1):67–81, 2005.

[14] J. Fiala, D. Paulusma, and J. A. Telle. Matrix and graph orders derived from locally constrained graph homomorphisms. In J. Jedrzejowicz and A. Szepietowski, editors, Mathematical Foundations of Computer Science 2005, 30th International Symposium, MFCS 2005, Gdansk, Poland, August 29 - September 2, 2005, Proceedings, volume 3618 of Lecture Notes in Computer Science, pages 340–351. Springer, 2005.

[15] F. V. Fomin, D. Lokshina, F. Panolan, S. Saurabh, and M. Zehavi. Finding, hitting and packing cycles in subexponential time on unit disk graphs. In I. Chatzigiannakis, P. Indyk, F. Kuhn, and A. Muscholl, editors, 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland, volume 80 of LIPIcs, pages 65:1–65:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[16] J. Fox and J. Pach. A separator theorem for string graphs and its applications. Combinatorics, Probability & Computing, 19(3):371–390, 2010.

[17] J. Fox and J. Pach. Computing the independence number of intersection graphs. In Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011, pages 1161–1165, 2011.

[18] D. Gonçalves, L. Isenmann, and C. Pennarun. Planar Graphs as L-intersection or L-contact graphs. In A. Czumaj, editor, Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018, pages 172–184. SIAM, 2018.

[19] J. R. Griggs and R. K. Yeh. Labelling graphs with a condition at distance 2. SIAM J. Discrete Math., 5(4):586–595, 1992.

[20] C. Groenland, K. Okrasa, P. Rzążewski, A. Scott, P. Seymour, and S. Spirkl. H-colouring P_t-free graphs in subexponential time. CoRR, abs/1803.05396v2, 2018.

[21] G. Z. Gutin, P. Hell, A. Rafiey, and A. Yeo. A dichotomy for minimum cost graph homomorphisms. Eur. J. Comb., 29(4):900–911, 2008.

[22] G. Hahn, J. Kratochvíl, J. Sirán, and D. Sotteau. On the injective chromatic number of graphs. Discrete Mathematics, 256(1-2):179–192, 2002.

[23] F. Havet, M. Klazar, J. Kratochvíl, D. Kratsch, and M. Liedloff. Exact algorithms for L(2, 1)-labeling of graphs. Algorithmica, 59(2):169–194, 2011.
[24] P. Hell and J. Nesetril. *Graphs and Homomorphisms*. Oxford University Press, jul 2004.

[25] P. Hell and A. Rafiey. The dichotomy of list homomorphisms for digraphs. In D. Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1703–1713. SIAM, 2011.

[26] P. Hell, A. Raspaud, and J. Stacho. On injective colourings of chordal graphs. In E. S. Laber, C. F. Bornstein, L. T. Nogueira, and L. Faria, editors, *LATIN 2008: Theoretical Informatics, 8th Latin American Symposium, Búzios, Brazil, April 7-11, 2008, Proceedings*, volume 4957 of *Lecture Notes in Computer Science*, pages 520–530. Springer, 2008.

[27] R. Impagliazzo and R. Paturi. On the complexity of $k$-SAT. *Journal of Computer and System Sciences*, 62(2):367 – 375, 2001.

[28] S. Kisfaludi-Bak and T. C. van der Zanden. On the exact complexity of Hamiltonian Cycle and $q$-Colouring in disk graphs. In D. Fotakis, A. Pagourtzis, and V. T. Paschos, editors, *Algorithms and Complexity - 10th International Conference, CIAC 2017, Athens, Greece, May 24-26, 2017, Proceedings*, volume 10236 of *Lecture Notes in Computer Science*, pages 369–380, 2017.

[29] P. Koebe. Kontaktprobleme der konformen Abbildung. *Berichte über die Verhandlungen der Sachsischen Akademie der Wissenschaften zu Leipzig, Mathematisch-Physikalische Klasse*, 88:141–164, 1936.

[30] J. Kratochvíl. String graphs. I. The number of critical nonstring graphs is infinite. *J. Comb. Theory, Ser. B*, 52(1):53–66, 1991.

[31] J. Kratochvíl. String graphs. II. Recognizing string graphs is NP-hard. *J. Comb. Theory, Ser. B*, 52(1):67–78, 1991.

[32] J. Kratochvíl and J. Matoušek. String graphs requiring exponential representations. *J. Comb. Theory, Ser. B*, 53(1):1–4, 1991.

[33] J. Kratochvíl and J. Matoušek. Intersection graphs of segments. *Journal of Combinatorial Theory, Series B*, 62(2):289 – 315, 1994.

[34] J. R. Lee. Separators in region intersection graphs. *CoRR*, abs/1608.01612, 2016.

[35] R. Lipton and R. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

[36] E. L. Lloyd and S. Ramanathan. On the complexity of distance-2 coloring. In *Proceedings ICCI ’92: Fourth International Conference on Computing and Information*, pages 71–74, May 1992.

[37] G. MacGillivray and J. Swarts. The complexity of locally injective homomorphisms. *Discrete Mathematics*, 310(20):2685–2696, 2010.

[38] D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. In N. Bansal and I. Finocchi, editors, *ESA 2015 Proc.*, volume 9294 of *LNCS*, pages 865–877. Springer, 2015.

[39] D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. *CoRR*, abs/1504.05476, 2015.

[40] J. Matoušek. Intersection graphs of segments and $\exists R$. *CoRR*, abs/1406.2636, 2014.

[41] J. Matoušek. Near-optimal separators in string graphs. *Combinatorics, Probability & Computing*, 23(1):135–139, 2014.

[42] C. McDiarmid and T. Müller. Integer realizations of disk and segment graphs. *J. Comb. Theory, Ser. B*, 103(1):114–143, 2013.

[43] G. L. Miller, S. Teng, W. P. Thurston, and S. A. Vavasis. Separators for sphere-packings and nearest neighbor graphs. *J. ACM*, 44(1):1–29, 1997.

[44] B. M. E. Moret. Planar NAE3SAT is in P. *SIGACT News*, 19(2):51–54, June 1988.
[45] M. Schaefer, E. Sedgwick, and D. Štefankovič. Recognizing string graphs in NP. *J. Comput. Syst. Sci.*, 67(2):365–380, 2003.

[46] M. Schaefer and D. Štefankovič. Fixed points, Nash equilibria, and the Existential Theory of the Reals. *Theory of Computing Systems*, 60(2):172–193, Feb 2017.

[47] W. D. Smith and N. C. Wormald. Geometric separator theorems. available online at [https://www.math.uwaterloo.ca/~nwormald/papers/focssep.ps.gz](https://www.math.uwaterloo.ca/~nwormald/papers/focssep.ps.gz).

[48] W. D. Smith and N. C. Wormald. Geometric separator theorems and applications. In *FOCS 1998 Proc.*, pages 232–243, Washington, DC, USA, 1998. IEEE Computer Society.

[49] J. van den Heuvel and S. McGuinness. Coloring the square of a planar graph. *Journal of Graph Theory*, 42(2):110–124, 2003.