



An 11/6-Approximation Algorithm for Vertex Cover on String Graphs

Édouard Bonnet   

Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Paweł Rzażewski  

Warsaw University of Technology & University of Warsaw, Warsaw, Poland

Abstract

We present a 1.8334-approximation algorithm for VERTEX COVER on string graphs given with a representation, which takes polynomial time in the size of the representation; the exact approximation factor is $\frac{11}{6}$. Recently, the barrier of 2 was broken by Lokshtanov et al. [SoGC '24] with a 1.9999-approximation algorithm. Thus we increase by three orders of magnitude the distance of the approximation ratio to the trivial bound of 2. Our algorithm is very simple. The intricacies reside in its analysis, where we mainly establish that string graphs without odd cycles of length at most 11 are 8-colorable. Previously, Chudnovsky, Scott, and Seymour [JCTB '21] showed that string graphs without odd cycles of length at most 7 are 80-colorable, and string graphs without odd cycles of length at most 5 have bounded chromatic number.

1 Introduction

The VERTEX COVER problem¹ is one of Karp's 21 NP-complete problems [11]. While it admits several easy polynomial-time 2-approximation algorithms, it is an open question if an approximation factor of $2 - \varepsilon$ can be achieved for some $\varepsilon > 0$. If the unique games conjecture (UGC) holds, then the answer to the latter question is negative [14]. Under the sole $P \neq NP$ assumption, it is currently only known that VERTEX COVER cannot be approximated within ratio less than $\sqrt{2}$ [8, 12, 13].

On several graphs classes, better approximation algorithms of VERTEX COVER exist. For instance, this problem can be exactly solved in polynomial-time on bipartite graphs, as it reduces to a maximum flow problem [15]. It also admits a polynomial-time approximation scheme (PTAS) on planar graphs by Baker's technique [2]. On the contrary, a $(2 - \varepsilon)$ -approximation algorithm for VERTEX COVER with $\varepsilon \in (0, 1/2]$ on triangle-free graphs would imply the same holding for general graphs. Indeed, one can observe that any vertex cover intersects each triangle on at least two vertices. One can thus repeatedly include all three vertices of a triangle in the approximate solution. Once the input graph has no triangle left, one calls the $(2 - \varepsilon)$ -approximation algorithm. This strategy can be seen to achieve approximation factor $\max(3/2, 2 - \varepsilon)$ on general graphs. In particular, the former algorithm on triangle-free graphs would refute the UGC. Thus we say that $(2 - \varepsilon)$ -approximating VERTEX COVER on triangle-free graphs is UGC-hard. It would be interesting to establish a dichotomy splitting the hereditary² graph classes on which we know a $(2 - \varepsilon)$ -approximation algorithm from those on which this task is UGC-hard.

The class of *string graphs* consists of every intersection graph of connected sets in a planar graph. Alternatively string graphs are the intersection graphs of (non-self-intersecting,

¹ As an optimization problem, given a graph G , find a smallest possible vertex subset S such that $G - S$ (i.e., the subgraph of G induced by all the vertices *not* in S) is edgeless.

² closed under vertex removals

non-closed) curves in the planes, called *strings*. The set of strings is called a *geometric representation* or *string representation*. The geometric representation may be convenient for drawing and in the proofs, but it is not so easy to handle as input. For that, we will also adopt a more combinatorial approach, based on the first definition of string graphs. A *representation* of a string graph G is a planar graph P and as many connected vertex subsets of P as G has vertices; two vertices are joined by an edge if and only if their corresponding subsets intersect. A caveat here is that finding such a representation is NP-hard [18]. Furthermore, some string graphs G require representations P whose number of vertices is exponential in $|V(G)|$ [19].

Recently Lokshtanov, Panolan, Saurabh, Xue, and Zehavi [23] presented a $(2 - \varepsilon)$ -approximation algorithm for VERTEX COVER on string graphs, for some small positive $\varepsilon \approx 10^{-4}$. In this paper, we give a simpler algorithm with improved approximation factor $11/6 = 1.8333\dots$

► **Theorem 1.** *VERTEX COVER admits an $\frac{11}{6}$ -approximation algorithm in string graphs given with a representation, whose running time is polynomial in the size of a representation.*

Our algorithm works as follows. We first get rid of all the odd cycles of length at most 11. This is done as in the abovementioned reduction to triangle-free graphs. While there is an odd cycle C with $|V(C)| \leq 11$, any vertex cover contains at least 6 vertices of C . We thus include all vertices of C in our approximate solution, and remove them from the graph. This ensures a 11/6 approximation factor, if such a factor can be achieved in the resulting graph. We can thus assume that our input graph has *odd girth*³ more than 11. As the standard linear-programming (LP) formulation of VERTEX COVER is half-integral [26], we can further assume that the vertex cover contains at least half of the vertices. Note that these opening steps are also present in the paper of Lokshtanov et al. [23].

We now deviate from the previous algorithm, and simply bound the chromatic number of string graphs of odd girth larger than 11.

► **Theorem 2.** *Every string graph of odd girth larger than 11 is 8-colorable, and given a representation, an 8-coloring can be found in time polynomial in the representation.*

A largest color class in a proper 8-coloring has size at least $n/8$ in an n -vertex graph. Thus, its complement is a vertex cover of size at most $7n/8$. As the optimum solution has size at least $n/2$, this yields a $7/4 = 1.75$ factor, and we conclude. Therefore, the main technical content of the paper is the proof of Theorem 2.

Outline of the proof of Theorem 2. Our strategy goes as follows. We make a breadth-first search (BFS) from some arbitrary vertex u_0 in each connected component of the graph G . We reserve colors 1, 2, 3, 4 for the even-indexed BFS layer, and 5, 6, 7, 8 for the odd-indexed BFS layer. We thus need to 4-color each connected component X of each subgraph induced by a single layer. Let R be a string representation of the entire graph, and $R[X]$ its restriction to X . As any face in the arrangement $R[X]$ can be made the infinite face, we can assume that the string of u_0 lies on the infinite face of $R[X]$. Now $R[X]$ has a nice property: The minimal topological disk D that contains it is such that each string of $R[X]$ intersects at least one string (in $R \setminus R[X]$) that itself crosses ∂D , the boundary of D . Indeed, such a string is given by a neighbor in the previous layer.

³ The (odd) girth of a graph G is the length of a shortest (odd) cycle of G .

We now consider the string representation R_H made by X and at least one neighbor of the previous layer for each vertex of X , and intersect it by D . Each string of X remains in one piece, whereas the strings of the previous layer are possibly split into several strings in D . Let H be the intersection graph of R_H . We further layer X with the distance in H to a fixed vertex $w \in V(H) \setminus V(X)$. Let X_k be the vertices of H at distance exactly k from w in H .

We are left with proving that $H[X_k]$ is bipartite. For the sake of contradiction, we assume that there is an odd cycle C in $H[X_k]$. Our goal is to show that this odd cycle is contained in a ball of small radius around some vertex, yielding a contradiction in the form of a short odd cycle. For that, we establish that there is another odd cycle C' (built from C) such that the string of w is contained in a face F made by few strings of $N_H[C']$, with most of the rest of C' not intersecting F . A string defining F is then at a small distance of *every* string of C' since a (shortest) path from w to a vertex of the rest of C' has to cross the boundary of F . After which, the path has only a constant number of steps to reach its target since vertices of $N_H[C']$ are at distance $k - 1$, k , or $k + 1$ from w . The short odd cycle in H implies the existence of a short odd cycle in G , a contradiction.

Chromatic number of intersection graphs without short (odd) cycles. The chromatic number of intersection graphs of a given girth has a long history. Erdős and Gyárfás asked if girth-4 (i.e., triangle-free) segment intersection graphs have bounded chromatic number [9]. Kostochka and Nešetřil [16] raised the same question for 1-string graphs.⁴ Both questions were answered in the negative by Pawlik et al. [27]. It was further shown by Walczak [29] that there are triangle-free segment intersection graphs without independent sets of size linear in the number of vertices. Nevertheless, Kostochka and Nešetřil [17] proved that 1-string graphs of girth at least 8 are 3-colorable, and asked [17, Problem 3] whether 8 could be replaced by 5. This has been confirmed for *outer* 1-strings by Das, Mukherjee, and Sahoo [6].

More relevant to our VERTEX COVER application, the chromatic number of (1-)string graphs of a given odd girth has also been considered. (Note indeed that, in the design of a $(2 - \varepsilon)$ -approximation algorithm for VERTEX COVER, whereas one can remove odd cycles up to any fixed size, the same trick does not apply to 4-vertex cycles.) McGuinness established that 1-string graphs of odd girth at least 7 have bounded chromatic number [24, 25]. Chudnovsky, Scott, and Seymour [5] further showed that string graphs of odd girth at least 9 are 80-colorable, and string graphs of odd girth at least 7 have bounded chromatic number. With proper adjustments, the former result can be turned into an algorithm that inputs a representation of any string graph G of odd girth at least 9, and outputs an 80-coloring of G in time polynomial in the representation. However, this would yield a significantly worse approximation factor for VERTEX COVER.

Limits of the method. The following observation summarizes the trade-off between required odd girth and effective upper bound on the chromatic number, in how they impact the approximation ratio.

► **Observation 3.** *Let \mathcal{C} be a class such that there is a polynomial-time algorithm to c -color the graphs of \mathcal{C} of odd girth at least an odd positive integer g . Then VERTEX COVER admits a polynomial-time $2 \max(\frac{g-2}{g-1}, \frac{c-1}{c})$ -approximation in \mathcal{C} .*

⁴ The 1-strings are strings every pair of which intersects at most once; note that it is not a property of particular objects but rather their arrangement.

We apply Observation 3 on represented string graphs with $g = 13$ and $c = 8$. Showing a counterpart of Theorem 2 with $g = 11$ and $c \leq 10$ would improve the ratio to $9/5 = 1.8$, and with $g = 9$ and $c \leq 8$, to $7/4 = 1.75$. Past this point, our 8-coloring starts being the bottleneck. The last possible step of this method would be to get parameters $g = 7$ and $c \leq 6$, for a ratio of $5/3 = 1.666\dots$. Indeed we recall that string graphs of odd girth at least 5 have unbounded chromatic number [27], and do not necessarily have linear-size independent sets [29]. On the complexity side, we do not expect that VERTEX COVER is APX-hard (i.e., NP-hard to approximate within some constant ratio $r > 1$) on represented string graphs as a quasipolynomial-time approximation scheme (QPTAS) exists [1, 10]. However, if a PTAS also exists, it will have to be found with a different approach than ours.

Let us emphasize that Theorem 2 is the only place where our algorithm actually requires a representation of the input to be given. Thus, an algorithm coloring string graphs with no short odd cycles, that works directly on the input graph G (not its representation), would yield an approximation algorithm for VERTEX COVER in string graphs whose complexity is polynomial in the number of vertices of G . However, if we are only interested in the decision variant of the problem, the existential statement in Theorem 2 is sufficient to get the following result.

► **Theorem 4.** *Given a graph G and an integer k , in time polynomial in $|V(G)|$ we can distinguish the following cases:*

1. G is a string graph and $\text{VERTEX COVER}(G) \leq k$, and
2. G is a string graph and $\text{VERTEX COVER}(G) > 11k/6$.

If none of the cases applies, the algorithm terminates but its output is unspecified.

Theorem 4 can be seen as a polynomial-time solution to the *promise variant* of VERTEX COVER, where the input is guaranteed to satisfy one of the above conditions; see [20, 28].

Let us remark that Lokshtanov et al. [23] claim that their algorithm can actually return a 1.9999-approximate solution in time polynomial in the number of vertices of the input graph but this is, unfortunately, not true. Indeed, one of the crucial steps in their argument is the application of a result of Lee [22, Theorem 4.2], whose proof has recently been realized to be flawed⁵ [3, 21]. Seemingly the result [22, Theorem 4.2] can be reproven in a different way but only for *region intersection graphs* (a generalization of string graphs) [7], and the new approach requires that the representation is given (and still the final approximation ratio is much closer to 2 than the one given by Theorem 1). Similarly, Lokshtanov et al. [23] claim that their approach applies for all proper *induced-minor-closed* classes, but the problem lies in the same place: we are not aware of any way to fix the proof of Lee [22, Theorem 4.2] in such a general setting.

We leave as open questions if VERTEX COVER admits, for some constant $\varepsilon > 0$, a $(2 - \varepsilon)$ -approximation algorithm on string graphs given without representation (which is very likely to be the case), and on classes excluding a fixed graph H as an induced minor.⁶

Further discussion and questions. An interesting notion to attack our question of which classes admit a less-than-2 approximation factor is that of r -controlledness (see [5]). For an integer $r \geq 1$, a class \mathcal{C} is r -controlled if there exists a function $f : \mathbb{N} \rightarrow \mathbb{N}$, such that for every $G \in \mathcal{C}$, the chromatic number of G is bounded by $f(\chi^r(G))$, where $\chi^r(G)$ is the maximum chromatic number of a ball $B^r(v)$ of radius r centered at a vertex v of G (i.e.,

⁵ The inequality at the sixth line of the proof of Lemma 4.14 need not hold.

⁶ i.e., H cannot be obtained from graphs of the class by removing vertices and contracting edges

the graph induced by the vertices at distance at most r from v). The function f is in the definition above is called a *binding function*.

In particular, every χ -bounded class \mathcal{C} , i.e., a class whose chromatic number χ is upper bounded by a function of its clique number ω , is 1-controlled. Indeed, if f is a χ -binding function for \mathcal{C} , we have $\chi^1(G) \geq \omega(G)$ and $\chi(G) \leq f(\omega(G))$ for every $G \in \mathcal{C}$. Thus $\chi(G) \leq f(\chi^1(G))$ (note that f can be chosen non-decreasing). We further say that \mathcal{C} is *effectively r -controlled* if there is a polynomial-time algorithm that turns any $\chi^r(G)$ -coloring of $B^r(v)$ for any $G \in \mathcal{C}$ and each $v \in V(G)$, into an $f(\chi^r(G))$ -coloring of G . Note that \mathcal{C} is r -controlled and $G \in \mathcal{C}$ has odd girth larger than $2r + 1$, then for every $v \in V(G)$, the subgraph induced by $B^r(v)$ is bipartite. Consequently, similarly to Observation 3, we obtain the following.

► **Observation 5.** *VERTEX COVER admits a polynomial-time $\max(\frac{2r+1}{r+1}, 2(1 - \frac{1}{f(2)}))$ -approximation algorithm on any effectively r -controlled class with binding function f .*

Being r -controlled for at least some r is quite an inclusive property among “structured” graph classes [5]. Thus we wonder the following.

► **Question 1.** *Which hereditary classes \mathcal{C} are not (effectively) r -controlled for any r , and yet VERTEX COVER admits a polynomial-time $(2 - \varepsilon)$ -approximation algorithm in \mathcal{C} , for some constant $\varepsilon > 0$?*

A candidate class for Question 1 is made by all graphs of girth at least 5. There is no r for which this class is r -controlled, and to our knowledge, no $(2 - \varepsilon)$ -approximation algorithm nor UGC-hardness proof.

As we already mentioned, intersection graphs of 1-strings with girth at least 8 are 3-chromatic [17]. Indeed, these graphs are 2-degenerate, i.e., (all their induced subgraphs) have a vertex of degree at most 2, which gives a recursive 3-coloring strategy. While string graphs of large odd girth are not d -degenerate for any d (they contain arbitrarily large bipartite complete graphs), we conjecture that they are nevertheless 3-colorable.

► **Conjecture 6.** *There is an integer k such that the class of string graphs of odd girth at least k is 3-chromatic.*

A stronger form of Conjecture 6 is that it holds for $k = 7$.

2 Definitions and preparatory lemmas

If $i \leq j$ are two non-negative integers, we denote by $[i, j]$ the set $\{i, i + 1, \dots, j - 1, j\}$, and $[i]$ is a short-hand for $[1, i]$.

Graphs and odd cycles We denote by $V(G)$ and $E(G)$ the set of vertices and edges of a graph G , respectively. A graph H is an *induced subgraph* (resp. *subgraph*) of a graph G if H can be obtained from G by vertex deletions (resp. by vertex and edge deletions). For $S \subseteq V(G)$, the *subgraph of G induced by S* , denoted $G[S]$, is obtained by removing from G all the vertices that are not in S . Then $G - S$ is a short-hand for $G[V(G) \setminus S]$. A set $X \subseteq V(G)$ is connected (in G) if $G[X]$ has a single connected component.

If C is a cycle, once a direction along the cycle is fixed, we may denote by $C[x \rightarrow y]$ the subpath of C from $x \in V(C)$ to $y \in V(C)$, turning in this direction.

► **Lemma 7.** *Let G be a graph that has an induced odd cycle C , and let $v \in V(G) \setminus V(C)$ have at least two neighbors in $V(C)$. Then there is a subpath of C that forms with v an induced odd cycle of G .*

Proof. Let v_1, \dots, v_h list the neighbors of v on C , starting at an arbitrary fixed vertex and turning in a fixed (arbitrary) direction. The cycles C_1, \dots, C_h , each with vertex set $\{v\} \cup C[v_i \rightarrow v_{i \pmod{h}+1}]$ for $i \in [h]$, respectively, have a combined number of edges of $|V(C)| + 2h$; an odd number since $|V(C)|$ is odd. Hence at least one of C_1, \dots, C_h has odd length. \blacktriangleleft

Let G be a graph and $v \in V(G)$. We denote by $N_G(v)$ and $N_G[v]$, the open, respectively closed, neighborhood of v in G . For a non-negative integer r , we denote by $N_G^{\leq r}(v)$ (resp. $N_G^{\overline{r}}(v)$) the set of vertices of G at distance at most r (resp. exactly r) from v . We may call $N_G^{\leq r}(v)$ the ball of radius r around v (in G). We will rely on the observation that if a possibly long odd cycle is contained in a ball of small radius, then there is also a short odd cycle (in this ball).

► **Lemma 8.** *Let G be a graph, $v \in V(G)$, and r be a positive integer. Suppose $G[N_G^{\leq r}(v)]$ contains an odd cycle C . Then G (even $G[N_G^{\leq r}(v)]$) has an odd cycle of length at most $2r + 1$. Furthermore, if $N_G^{\overline{r}}(v) \cap V(C)$ is an independent set of G , then G (even $G[N_G^{\leq r-1}(v)]$) has an odd cycle of length at most $2r - 1$.*

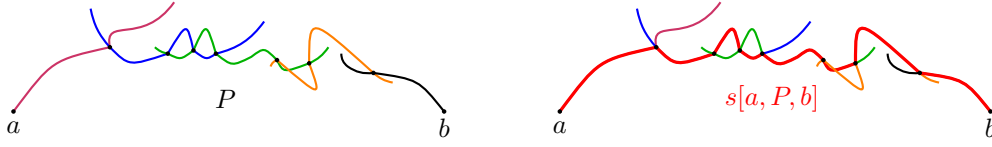
Proof. Perform a breadth-first search (BFS) starting at v in $G[N_G^{\leq r}(v)]$. By definition, this breadth-first search has exactly $r + 1$ layers L_0, L_1, \dots, L_r , with $L_i := N_G^{\overline{i}}(v)$ for each $i \in [0, r]$. At least one L_i is *not* an independent set, otherwise $G[N_G^{\leq r}(v)]$ would be bipartite, and would not contain any odd cycle. Say $x, y \in L_i$ are adjacent, and let $z \in L_{i'}$ be their lowest common ancestor in the BFS tree (possibly $z = v$ and $i' = 0$). Then, there is a cycle of length $2(i - i') + 1 \leq 2r + 1$ in $G[N_G^{\leq r}(v)]$, consisting of the edge xy and the paths from z to x and from z to y in the BFS tree. Furthermore, if $N_G^{\overline{r}}(v) \cap V(C)$ is an independent set of G , then the odd cycle C has at least one of its edges with both endpoints in some $L_i \neq L_r$, which makes an odd cycle of length at most $2r - 1$. \blacktriangleleft

String representations, and some useful notions and properties A *string* is a subset of \mathbb{R}^2 homeomorphic to a closed segment. We may call *substring* of a string s a path-connected subset of s . If R is a collection of strings in the plane, we denote by G_R their intersection graph, with one vertex per string, and an edge between every pair of intersecting strings. For any vertex $v \in V(G_R)$, we denote by $s_R(v)$ the string *representing* v in R . Conversely, we may denote by $v_R(s)$ the vertex represented by the string s . We typically drop the subscript when R is clear from the context. If X is an induced subgraph of G_R , we may denote by $R[X]$ the subset of strings of R corresponding to vertices of X .

A *face* of a geometric arrangement made by a collection \mathcal{S} of strings, or more generally of curves, in \mathbb{R}^2 is a connected component of $\mathbb{R}^2 \setminus \bigcup_{s \in \mathcal{S}} s$. A *closed face* is the topological closure of a face. If D is a topological disk, and more generally a face, we denote by ∂D its boundary. For instance, the curve ∂D defines two faces, one of which is finite ($D \setminus \partial D$), and admits D as a closed face. We may denote by \overline{F} the closure of an (open) face F .

If P is an induced path represented by strings s_1, \dots, s_h (i.e., s_i and s_j intersect whenever $|i - j| = 1$), $a \in s_1$ and $b \in s_h$, we denote by $s[a, P, b]$ a minimal path-connected subset of $\bigcup_{i \in [h]} s_i$ containing a and b . In particular $s[a, P, b]$ is a string with endpoints a and b , coinciding with each s_i on a substring of positive length. Although $s[a, P, b]$ is not necessarily uniquely defined, every further claim will hold regardless of the actual choice for $s[a, P, b]$. It can thus be thought as a short-hand for: *fix any minimal path-connected subset of $\bigcup_{i \in [h]} s_i$ containing a and b , and denote it $s[a, P, b]$.*

It will be convenient to extend the latter notation to induced cycles, that is, to allow s_1 and s_h to be adjacent. If P may be a cycle, then $s[a, P, b]$ is defined more specifically in



■ **Figure 1** Left: 5-vertex path as the intersection graph of strings, where the first string of the path contains point a , and the last string of the path contains point b . Right: illustration of $s[a, P, b]$.

the case when s_1 and s_h indeed intersect. If there is a point $c \in s_1 \cap s_h$ and two substrings $s'_1 \subseteq s_1, s'_h \subseteq s_h$ with endpoints a, c and b, c , respectively, such that $(s'_1 \cup s'_h) \cap \bigcup_{2 \leq i \leq h-1} s_i = \emptyset$, then $s[a, P, b] := s'_1 \cup s'_h$. If not, then there is a (*non-self-intersecting*) string that starts at a , follows non-empty substrings of s_1, \dots, s_h in this order, and ends at b . We fix, as before, $s[a, P, b]$ to be any such string.

We will need the following lemma.

► **Lemma 9.** *Let D be a topological disk in the plane. Let $V \uplus W$ be a set of strings all contained in D , whose intersection graph is a cycle C , such that*

- $|W| \geq 2$,
- every string of W has one or both endpoints in ∂D ,
- no other point of a string of $V \cup W$ intersects ∂D , and
- no two strings of $(V \cup W)$ intersect at a point of ∂D .

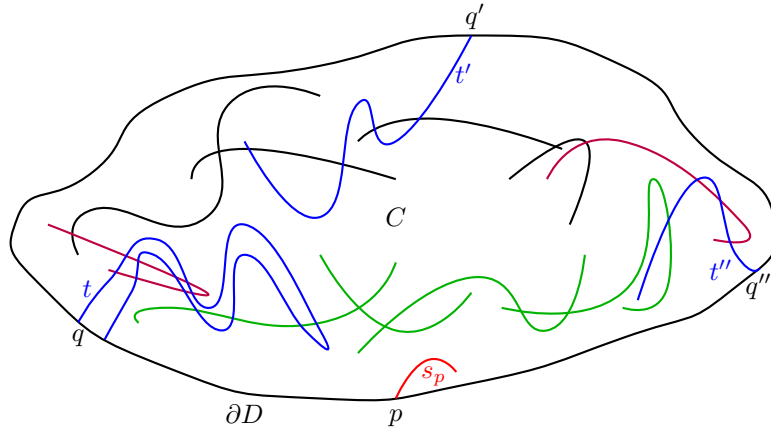
Let s_p be a string contained in D with at least one endpoint p in ∂D , such that s_p does not intersect any string of $V \cup W$.

Then there are two strings $s \neq s' \in W$, and h strings $s_1, \dots, s_h \in V$ with $v(s), v(s_1), \dots, v(s_h), v(s')$ a subpath of C (or C itself), and s_p is contained in a closed face \overline{F} of the arrangement $\{\partial D, s, s_1, \dots, s_h, s'\}$ such that \overline{F} does not intersect any string of $V \cup W$ outside of s, s_1, \dots, s_h, s' , the string $s_{-1} \neq s_1 \in V \cup W$ intersecting s , and the string $s_{h+2} \neq s_h \in V \cup W$ intersecting s' .

Proof. Let t, t' be two distinct strings of W , with $q \in t \cap \partial D$ and $q' \in t' \cap \partial D$. Let P_1, P_2 be the two paths from $v(t)$ to $v(t')$ in C . We define the strings $t_1 := s[q, P_1, q']$ and $t_2 := s[q, P_2, q']$. We denote by $\langle q, q' \rangle$ and $\langle q', q \rangle$ the two minimal path-connected subsets of ∂D containing q and q' . Let F_1, F'_1 be the two finite (open) faces with boundary $\langle q, q' \rangle \cup t_1$ and $\langle q', q \rangle \cup t_1$, respectively. Similarly let F_2, F'_2 be the two finite faces with boundary $\langle q, q' \rangle \cup t_2$ and $\langle q', q \rangle \cup t_2$, respectively.

As C is an induced cycle, at most one of $\overline{F_1}, \overline{F'_1}$ intersects the string of some vertex in $V(C) \setminus N_C[P_1]$. Without loss of generality, suppose that $\overline{F_1}$ does not intersect any string of $V(C) \setminus N_C[P_1]$. As s_p does not intersect t_1 , it is contained in the closed face $\overline{F_1}$ or in the closed face $\overline{F'_1}$. If s_p is contained in $\overline{F_1}$, we set $P := P_1 - \{v(t), v(t')\}$ and $F := F_1$, and proceed to the next paragraph. Otherwise, we observe that s_p is contained in $\overline{F'_2}$, and $\overline{F'_2}$ does not intersect any string of $V(C) \setminus N_C[P_2]$. In which case we set $P := P_2 - \{v(t), v(t')\}$ and $F := F'_2$.

At this point, we have all the requirements of the lemma except that some strings of P may be in W . Let P be v_1, \dots, v_ℓ with v_1 a neighbor of $v(t)$, and v_ℓ a neighbor of $v(t')$. While there is some string $t'' \in W$ with $v(t'') = v_i \in V(P)$, we let $q'' \in t'' \cap \partial D$ and distinguish two cases. If q, p, q'' turn along ∂D in the same direction as q, p, q' , we set $t' := t''$ and $P := v_1, \dots, v_{i-1}$. Otherwise we set $t := t''$ and $P := v_{i+1}, \dots, v_\ell$. (Note that $|V(P)|$ decreases by at least one in each case.) When we exit the while loop, $s := t, s' := t' \in W$, and the subpath P of C (now only made of strings of V) satisfy the lemma statement. A visual



■ **Figure 2** Illustration of the proof of Lemma 9. The strings of W are in blue. Say, we initially pick $t \neq t' \in W$ as depicted. The $v(t)-v(t')$ subpath of C enclosing s_p contains a string in W , namely t'' . As qpq'' turns as qpq' along ∂D , t' will be set to t'' , and the new P (green strings) is entirely in V , so the process stops. The strings $s := t, s' := t''$, the green strings, and ∂D form a closed face containing s_p that, among the strings of the rest of C , only the two purple strings (may) intersect.

recap is provided by Figure 2. ◀

3 String graphs of odd girth larger than 11 are 8-colorable

This section is devoted to the proof of Theorem 2. Let G be a string graph of odd girth strictly greater than 11, and R be a string representation of G . We may color each connected component of G independently, thus assume that G is connected. Let $u_0 \in V(G)$ be an arbitrary vertex. For each $i \geq 0$, let $L_i := N_G^{\leq i}(u_0) \subseteq V(G)$ be the set of vertices at distance exactly i from u_0 . Note that $\{u_0\} = L_0, L_1, L_2, \dots$ partition $V(G)$, and that there may be an edge between $x \in L_i$ and $y \in L_j$ only if $|i - j| \leq 1$.

Our goal is to 4-color $G[L_i]$ for each non-negative integer i . We are then done, since we can use two disjoint 4-color palettes for $G[\bigcup_{i \text{ is odd}} L_i]$ and $G[\bigcup_{i \text{ is even}} L_i]$. We fix i , and assume that $i \geq 2$ since L_0 is a singleton, and L_1 is an independent set, due to the condition on the odd girth. Note that it is sufficient to 4-color each connected component of $G[L_i]$. Let X be the vertex set of any connected component of $G[L_i]$.

Definition of the topological disk D . Let $R[X]$ be the string representation of $G[X]$ obtained by keeping the strings of R corresponding to vertices of X . Let D_0 be the topological disk whose boundary is the boundary of the infinite face in the arrangement $R[X]$. If $s(u_0)$ is contained in D_0 , we draw R on a sphere, place any free point (i.e., not occupied by a string of R) of the face containing $s(u_0)$ in $R[X]$ at its north pole, and consider the stereographic projection of this string representation. The latter is such that $s(u_0)$ is on the infinite face of the string arrangement $R[X]$. Thus we can in fact assume that $s(u_0)$ is not contained in D_0 .

Note that every string of $R[X]$ is contained in D_0 , and apart from those of $L_{i-1} \cup L_{i+1} \cup X$ no string of R intersects ∂D_0 . Let $D \supset D_0$ be a very slightly augmented topological disk such that $\partial D_0 \subset D \setminus \partial D$ and the property that no string outside $L_{i-1} \cup L_{i+1} \cup X$ intersects ∂D is preserved. We can also ensure that no intersection of two strings of R lies on ∂D , and, for any point $q \in \partial D$, no string of R intersects ∂D at q without crossing it at q . We observe

that every string of L_{i-1} with a neighbor in $V(X)$ intersects ∂D , and that every vertex of X has a neighbor in L_{i-1} .

Definition of the auxiliary graph H . Let R_H be the set of strings formed by $D \cap R[(L_{i-1} \cap N_G(V(X))) \cup V(X)]$, and H its intersection graph. Every vertex of X corresponds to a single vertex in H , whereas each vertex of $L_{i-1} \cap N_G(V(X))$ may split in several strings in H as the corresponding string in R may enter and exit D several times. However, the strings of R_H that correspond to the same vertex in $L_{i-1} \cap N_G(V(X))$ are pairwise non-intersecting and thus they form an independent set in H . Fix an arbitrary vertex $w \in L_{i-1} \cap N_G(V(X))$. Let X_j be the subset of vertices of X at distance exactly j from w in H . Again, X_1, X_2, \dots partition X . By the previous remarks, it is sufficient to show that for any positive integer k , the graph $G[X_k] = H[X_k]$ is bipartite. We fix k , and show this fact in the next section.

3.1 $G[X_k]$ is bipartite

For the sake of contradiction assume that there is an induced odd cycle C in $G[X_k]$. We denote by v_1, \dots, v_h the vertices of C . For every $\ell \in [h]$, we fix some $w_\ell \in N_H(v_\ell) \setminus X$. Such a vertex exists since every vertex of L_i has at least one neighbor in L_{i-1} (possibly split into several strings in R_H , at least one of which intersects $s(v_\ell)$). Observe that w_1, \dots, w_h are not necessarily pairwise distinct. We set $W := \{w_1, \dots, w_h\}$, with $1 \leq |W| \leq h$. We say that string $s(v_i)$ (and, by extension, vertex v_i) is *simply attached* in a cycle C' containing v_i if w_i has only one neighbor in $V(C')$, namely v_i . We denote by $r(w_i)$ one (arbitrary, if both endpoints are in ∂D) endpoint of w_i that is in ∂D .

Our plan is to show that there is an odd cycle C' contained in the ball of radius 6 around some vertex z in H such that $N_H^{\leq 6}(z) \cap V(C')$ is an independent set. This implies, by Lemma 8, the existence of an odd cycle of length at most 11 in H , which in turn implies, as we will see, that the same happens in G . To do this, we exhibit a path \hat{P} in $H[V(C) \cup W]$ on at most four vertices, whose strings define with ∂D a (finite) closed face F containing $s(w)$ (recall that w is the arbitrary BFS root in H), and an odd cycle C' in $H[V(C) \cup W]$ “mostly” contained in $D \setminus F$. We then show that z can be chosen among the vertices of \hat{P} .

Let us first establish the following lemma.

► **Lemma 10.** *There is an induced odd cycle C' in H such that $V(C') \subseteq V(C) \cup W$, such that one of the following items holds*

- $C' = C$ and every string of $V(C')$ is simply attached in C' , or
- $V(C') \cap W = \{w_i\}$, for some $w_i \in W$, and $V(C') \setminus \{w_i\}$ induces a subpath of C whose internal vertices are all simply attached in C' , or
- $|V(C') \cap W| \geq 2$, and there is a subpath x, v_i, \dots, v_j, y of C' (or C) such that $x \neq y \in W$, $\{v_i, \dots, v_j\} \subset V(C)$ may be empty, $s(v_{i+1}), \dots, s(v_{j-1})$ are all simply attached in C' , and $s(w)$ is contained in a closed face of $\{\partial D, s(x), s(v_i), s(v_{i+1}), \dots, s(v_{j-1}), s(v_j), s(y)\}$ that does not intersect any string of C' but $s(x), s(v_i), s(v_{i+1}), \dots, s(v_{j-1}), s(v_j), s(y)$ and the other neighbor in C' of x and of y .

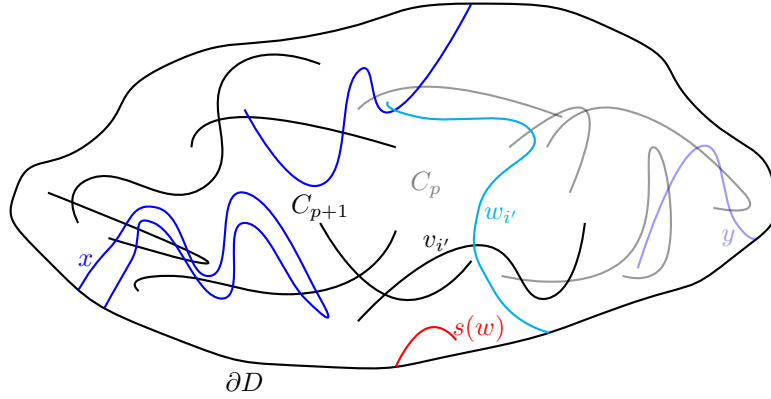
Proof. If all the strings of C are simply attached, we are done as the first item holds. Otherwise there is some $w_i \in W$ with at least two neighbors in $V(C)$. By Lemma 7, there is an induced odd cycle C_1 such that $V(C_1)$ comprises w_i and a subpath of C .

Let $p \geq 1$, and suppose we have defined C_p . If C_p satisfies the second or the third item of the lemma, we are done; so we assume otherwise. We will show that we can find an induced cycle C_{p+1} in $H[V(C) \cup W]$ with fewer vertices in $V(C)$ than C_p has.

First, suppose that $V(C_p) \cap W$ is a singleton $\{x\}$. As C_p does not satisfy the second outcome, there is a vertex $v_i \in V(C_p) \setminus \{x\} \subseteq V(C)$ that is *not* simply attached nor adjacent to x in C_p . We then obtain C_{p+1} by applying Lemma 7 to the pair C_p, w_i .

Suppose instead that C_p has at least two vertices in W . By Lemma 9, $s(w)$ is contained in a closed face F of the arrangement $\{\partial D, s(x), s(v_i), s(v_{i+1}), \dots, s(v_{j-1}), s(v_j), s(y)\}$ with $x \neq y \in W$ such that F does not intersect any string of C' outside $s(x'), s(x), s(v_i), s(v_{i+1}), \dots, s(v_{j-1}), s(v_j), s(y), s(y')$ where x', y' are the unique vertices in $N_{C'}(x) \setminus \{v_i\}, N_{C'}(y) \setminus \{v_j\}$, respectively. As C_p does not satisfy the third item, $|j - i| \geq 2$ and there is some $v_{i'}$ with $i + 1 \leq i' \leq j - 1$ such that $s(v_{i'})$ is *not* simply attached in C_p . We then obtain C_{p+1} by applying Lemma 7 to the pair $C_p, w_{i'}$; see Figure 3.

In both cases, the number of vertices of $V(C)$ present in the current odd cycle drops by at least one when going from C_p to C_{p+1} . Thus, after at most $h = |V(C)|$ iterations, the procedure will return an odd cycle C_q in $H[V(C) \cup W]$ that satisfies the second or the third statement of the lemma. \blacktriangleleft



■ **Figure 3** The string $w_{i'} \in W$ with several neighbors in $V(C_p)$, and the new odd induced cycle C_{p+1} obtained by Lemma 7. Lemma 9 will then locate $s(w)$ as enclosed by $x, w_{i'}$ and some (here, three) strings of V . For legibility, a string may be labeled by its corresponding vertex.

We can then obtain the announced milestone.

► **Lemma 11.** *There is an induced odd cycle C' in $H[V(C) \cup W]$, two vertices $x \neq y \in W$, possibly part of C' , and a subpath P of C' on 0, 1, or 2 vertices, all in $V(C)$, such that*

- *if $V(P) \neq \emptyset$, then x is adjacent to one endpoint of P , and y , to the other (possibly equal) endpoint of P , and if $V(P) = \emptyset$, then x and y are adjacent,*
- *one finite closed face F_1 of the arrangement $\{\partial D, s[r(x), xPy, r(y)]\}$ contains $s(w)$, and*
- *the other finite closed face F_2 contains every string of $V(C') \setminus N_{C'}[V(P) \cup \{x, y\}]$.*

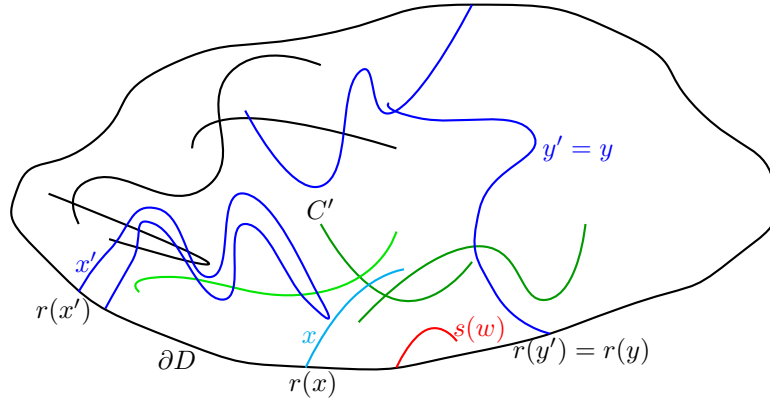
Proof. By Lemma 10, there is an induced odd cycle C' in $H[V(C) \cup W]$, two vertices $x' \neq y' \in W$, possibly part of C' , and a possibly-empty subpath P of C' such that

- $V(P) \subset V(C)$ and each internal vertex of P is simply attached in C' ,
- $x'Py'$ is a path or a cycle (we allow x' and y' to be adjacent, even when $V(P) \neq \emptyset$),
- $\{\partial D, s[r(x'), x'Py', r(y')]\}$ has two finite closed faces $F_1 \supset s(w)$, and
- F_2 containing every string of $V(C') \setminus N_{C'}[V(P) \cup \{x', y'\}]$.

Indeed, we directly get this outcome if the third item of Lemma 10 holds. If instead the first item of Lemma 10 holds, we set $x' := w_i$ and $y' := w_j$ for any $v_i \neq v_j \in V(C) = V(C')$ such that $w_i w_j \notin E(H)$. This is possible to ensure since H is triangle-free and

$|V(C)| \geq 3$. The path P is then chosen as the subpath of C from v_i to v_j such that $\partial D \cup s(x') \cup \bigcup_{u \in V(P)} s(u) \cup s(y')$ separates $s(w)$ from $V(C) \setminus N_C[P]$. If finally the second item of Lemma 10 holds, we set $x' := w_i$ for any $v_i \in V(C')$ such that $w_i \notin V(C')$, and y' is defined as the only vertex of $W \cap V(C')$. We then set P as previously.

Now, while the path P has at least three vertices and $s[r(x'), x'Py', r(y')]$ passes through the strings of P , let v_i be an internal vertex of P . By construction, $s(v_i)$ is simply attached in C' . Let P' be the subpath of P going from the endpoint of P neighboring x' to v_i , and P'' be the subpath of P going from v_i to its other endpoint (neighboring y'). Then, either P' and the pair $x' \neq w_i \in W$ or P'' and the pair $w_i \neq y'$ satisfy the four items of the previous paragraph. In the former case, we set $P := P'$ and $y' := w_i$, whereas in the latter, we set $P := P''$ and $x' := w_i$; see Figure 4.



■ **Figure 4** Example of strings $x' \neq y' \in W$ and the subpath P (three green strings) of C' . The internal node of P is simply attached to C' . This is the case when x' should be updated (to the vertex of the cyan string). The new path P (darker green) has two vertices, thus the process stops.

We are done when $|V(P)| \leq 2$ or when $s[r(x'), x'Py', r(y')]$ is contained in $s(x') \cup s(y')$. In both cases, we set $x := x'$ and $y := y'$, and in the latter case, we set P to be empty. ◀

Following the notations of Lemma 11, we define z as the neighbor of x in P if $V(P) \neq \emptyset$, and as x otherwise.

► **Lemma 12.** $V(C') \subseteq N_H^{\leq 6}(z)$.

Proof. We keep the notations of Lemma 11. First we observe that any vertex in $N_{C'}[V(P) \cup \{x, y\}]$ is at distance at most 3 of z in H . It remains to see that every vertex z' of C' such that $s(z')$ is contained in F_2 is at distance at most 6 from z in H .

Note that every vertex of W is at distance $k - 1$, k , or $k + 1$ from w in H , since every vertex of C is at distance exactly k from w in H , and every vertex of W has a neighbor in $V(C)$. As $s(w)$ is in F_1 , a (shortest) path from w to z' has to contain a vertex y' such that $s(y')$ intersects $s[r(x), xPy, r(y)]$. Since vertices of $V(P) \cup \{x, y\}$ are at distance at least $k - 1$ from w , vertex y' is at distance at least $k - 2$ from w . Now, as vertices of C' are at distance at most $k + 1$ from w , vertex y' is at distance at most 3 from z' . In turn, z is at distance at most 3 from y' , and we conclude. ◀

We further show that no edge of C' can lie within $N_H^{\leq 6}(z)$.

► **Lemma 13.** $N_H^{\leq 6}(z) \cap V(C')$ is an independent set of H or H admits an odd cycle of length at most 9.

Proof. The proof of Lemma 12 shows that for some vertex $z' \in V(C')$ to be at distance 6 from z in H , there should be a 4-edge path from y to z' . Thus two vertices z', z'' adjacent in C' and at distance 6 from z in H entail an odd cycle of length *at most* 9 in H , built from y , the corresponding two (non-necessarily edge-disjoint) 4-edge paths, z' , and z'' . ◀

Lemmas 8, 12, and 13 imply the existence of an odd cycle C_o of length at most 11 in H . We finally see that this yields a contradiction, as we can build an odd cycle of at most the same length in G .

► **Lemma 14.** *Let C_o be an odd cycle in H . Then G has an odd cycle of length at most $|V(C_o)|$.*

Proof. We initialize a represented odd cycle \widehat{C} to C_o realized by the corresponding strings of R_H . We make an induction on the number of strings of \widehat{C} not part of the representation R of G . When this number is 0, we conclude since G contains the cycle \widehat{C} as a subgraph. Let $y_1, \dots, y_q \in V(\widehat{C})$ be all the vertices of \widehat{C} whose strings are substrings of $s(y)$ for some $y \in L_{i-1} \cap N_G(V(X))$.

We further assume that starting at $y_1 \in V(\widehat{C})$, and turning in some fixed (arbitrary) direction along \widehat{C} , one encounters y_1, y_2, \dots, y_q in this order. Let, for every $a \in [q]$, ℓ_a be the number of edges of $\widehat{C}[y_a \rightarrow y_{a \pmod{q}+1}]$. As, $|V(\widehat{C})| = \sum_{a \in [q]} \ell_a$, at least one ℓ_a is odd. Recall that $\{y_1, \dots, y_q\}$ is an independent set in H , hence every ℓ_a is strictly greater than 1. Then the string $s(y)$ and those of the internal vertices in the path between y_a and $y_{a \pmod{q}+1}$ form an odd cycle of length at least 3 and at most $|V(\widehat{C})|$. This defines the new represented odd cycle \widehat{C} , and concludes the proof. ◀

3.2 8-coloring algorithm

To claim Theorem 2, we finally need to check that, given a representation P, \mathcal{S} of a string graph G of odd girth larger than 11, one can compute an 8-coloring of G in time polynomial in $|V(P)|$. Recall that P is a planar graph and \mathcal{S} is a set of non-empty connected sets in P in one-to-one correspondence with $V(G)$ such that two distinct connected sets of \mathcal{S} intersect if and only if the corresponding vertices of G are adjacent. Note that, as G is in particular triangle-free, $|V(G)| \leq 2|V(P)|$.

We remind the reader that we compute one BFS in G , and less than $|V(G)|$ BFSes in auxiliary graphs H of size at most $(|V(P)| + 1)|V(G)|$. After this we simply 2-color bipartite graphs whose combined number of vertices is at most $|V(G)|$. Thus, we shall just detail how to compute each auxiliary graph H .

Let X be the connected component of $G[L_i]$ giving rise to H . We start by adding every vertex of X to $V(H)$. We compute the set $Y \subseteq V(P)$ of all the vertices contained in an element of \mathcal{S} corresponding to a vertex of X . Let $u' \in V(P)$ be a vertex in the connected set of u_0 . Let $Y' \subseteq Y$ be the vertices $v \in Y$ for which there is a path in P from u' to v whose internal vertices are all in $V(P) \setminus Y$. Note that Y' can be computed in polynomial time in $|V(P)|$ by checking if u' and y are in the same connected component of $P - (Y \setminus \{v\})$. The set Y' is the combinatorial counterpart of ∂D . In particular, we do not need to change the representation if u' is in a finite face “made by Y .” This was merely helpful in the proof correctness.

For each vertex $v \in L_{i-1} \cap N_G(V(X))$ (alternatively we can only keep at least one neighbor per vertex of X), we add to $V(H)$ one vertex for each connected component of $S \cap Y$ in P , where S is the connected set of v . Note that $S' \cap Y' \neq \emptyset$ for every vertex set S' of such a connected component. This step adds to H fewer than $|V(G)| \cdot |V(P)|$ vertices.

Graph H is finally defined as the intersection graph of all the connected sets of vertices in $V(H)$, which can be computed in time polynomial in $|V(P)|$.

4 Approximation algorithm for Vertex Cover

For a graph G , let $\text{vc}(G)$ denote the size of a minimum vertex cover in G . We will use the following result of Chlebík and Chlebíková [4], which is a slight strengthening of the well-known Nemhauser–Trotter theorem [26].

► **Theorem 15** (Chlebík and Chlebíková [4], Nemhauser and Trotter [26]). *Given a graph G , one can compute in polynomial time a partition of $V(G)$ into $V_0, V_{1/2}, V_1$, such that*

1. *there are no edges between V_0 and $V_{1/2}$ or within V_0 ,*
2. *$\text{vc}(G[V_{1/2}]) \geq \frac{1}{2}|V_{1/2}|$, and*
3. *every minimum vertex cover S of G satisfies $V_1 \subseteq S \subseteq V_1 \cup V_{1/2}$.*

Finally, we are ready to prove Theorem 1, which we restate for convenience.

► **Theorem 1.** *VERTEX COVER admits an $\frac{11}{6}$ -approximation algorithm in string graphs given with a representation, whose running time is polynomial in the size of a representation.*

Proof. Let G be the input graph, given along with a representation. The algorithm has three phases.

Phase one. We initialize $X = \emptyset$. If G contains an odd cycle of length at most 11, we include all its vertices into X and remove them from the graph. We repeat this step exhaustively; clearly this can be done in polynomial time. Let G' be the graph obtained after the last iteration of the process.

Phase two. We call the algorithm from Theorem 15 on the graph G' in order to obtain three sets $V_0, V_{1/2}$, and V_1 . We denote $Y = V_1$ and $G'' = G'[V_{1/2}]$. Note that G'' is a string graph with odd girth larger than 11 and the representation of G'' can be easily obtained from the representation of G by removing strings (or connected sets) corresponding to vertices in $V(G) \setminus V(G'')$.

Phase three. We apply the algorithm from Theorem 2 to find a proper coloring of G'' with at most 8 colors. Let c be the color that appears most frequently, and let Z be set of vertices of G'' not colored c . Clearly $|Z| \leq \frac{7}{8}|V(G'')|$, so $|V(G'')| \geq \frac{8}{7}|Z|$. The algorithm returns $Q = X \cup Y \cup Z$.

Analysis. First, let us argue that Q is indeed a vertex cover. Since $X \subseteq Q$, it is enough to show that $Q \cap V(G') = Y \cup Z$ is a vertex cover of G' . Note that by the first property in Theorem 15 and since $V_1 = Y$, all edges of G' not contained in G'' are covered by Q . So we are left with showing that $Q \cap V(G'') = Z$ is a vertex cover of G'' . This is clearly true, as the complement of Z in G'' is an independent set (of color c).

Now let us analyze the approximation factor. Let S be an optimum solution, i.e., a vertex cover of G of size $\text{vc}(G)$. Note that for each odd cycle C removed in the first phase, $S \cap C$ must contain at least $\frac{|C|}{2}$ vertices in order to cover all the edges of C . As each removed cycle has at most 11 vertices, we conclude that $|S \cap X| \geq \frac{6}{11}|X|$.

Note that $S' \setminus X$ is a vertex cover of G' . By the third property in Theorem 15 we observe that $Y = V_1 \subseteq S$, and so $S \setminus (X \cup Y)$ is a vertex cover of G'' . By the second property in Theorem 15 we obtain that $|S \setminus (X \cup Y)| \geq \frac{1}{2}|V(G'')|$.

Summing up, we obtain

$$\begin{aligned} |S| &= |S \cap X| + |S \cap Y| + |S \setminus (X \cup Y)| \geq \frac{6}{11}|X| + |Y| + \frac{1}{2}|V(G'')| \\ &\geq \frac{6}{11}|X| + |Y| + \frac{4}{7}|Z| \geq \frac{6}{11}|X \cup Y \cup Z| = \frac{6}{11}|Q|, \end{aligned}$$

which means that $|Q| \leq \frac{11}{6}|S| = \frac{11}{6}\text{vc}(G)$. This completes the proof. \blacktriangleleft

The proof above is easily adapted to show Theorem 4. The first two phases remain unchanged. Let \mathcal{C} be the family of odd cycles found in phase one and let Y be the set found in phase two. Once we reach phase three, we do not call the coloring algorithm on G'' , as its running time might not be polynomial in $|V(G)|$. Instead, we check if $\sum_{C \in \mathcal{C}} [|V(C)|/2] + |Y| + |V(G'')|/2 > k$ and, if so, we reject the instance. Note that the sum in the expression above is a lower bound on $\text{vc}(G)$, so this step is correct. Otherwise, we accept the instance. The approximation guarantee is estimated as in the proof of Theorem 1.

References

- 1 Anna Adamaszek, Sariel Har-Peled, and Andreas Wiese. Approximation schemes for independent set and sparse subsets of polygons. *J. ACM*, 66(4):29:1–29:40, 2019. doi:10.1145/3326122.
- 2 Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994. doi:10.1145/174644.174650.
- 3 Édouard Bonnet, Jędrzej Hodor, Tuukka Korhonen, and Tomáš Masařík. Treewidth is polynomial in maximum degree on weakly sparse graphs excluding a planar induced minor. *CoRR*, abs/2312.07962, 2023. URL: <https://doi.org/10.48550/arXiv.2312.07962>, arXiv:2312.07962, doi:10.48550/ARXIV.2312.07962.
- 4 Miroslav Chlebík and Janka Chlebíková. Improvement of Nemhauser-Trotter theorem and its applications in parametrized complexity. In Torben Hagerup and Jyrki Katajainen, editors, *Algorithm Theory - SWAT 2004, 9th Scandinavian Workshop on Algorithm Theory, Humlebaek, Denmark, July 8-10, 2004, Proceedings*, volume 3111 of *Lecture Notes in Computer Science*, pages 174–186. Springer, 2004. doi:10.1007/978-3-540-27810-8_16.
- 5 Maria Chudnovsky, Alex Scott, and Paul D. Seymour. Induced subgraphs of graphs with large chromatic number. v. chandeliers and strings. *J. Comb. Theory B*, 150:195–243, 2021. URL: <https://doi.org/10.1016/j.jctb.2021.05.001>, doi:10.1016/J.JCTB.2021.05.001.
- 6 Sandip Das, Joydeep Mukherjee, and Uma kant Sahoo. Outer 1-string graphs of girth at least five are 3-colorable. In *Extended Abstracts EuroComb 2021: European Conference on Combinatorics, Graph Theory and Applications*, pages 593–598. Springer, 2021.
- 7 James Davies. private communication.
- 8 Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. Towards a proof of the 2-to-1 games conjecture? In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 376–389. ACM, 2018. doi:10.1145/3188745.3188804.
- 9 András Gyárfás. Problems from the world surrounding perfect graphs. *Applicationes Mathematicae*, 19(3-4):413–441, 1987.

- 10 Sarel Har-Peled. Approximately: Independence implies vertex cover. *CoRR*, abs/2308.00840, 2023. URL: <https://doi.org/10.48550/arXiv.2308.00840>, arXiv: 2308.00840, doi:10.48550/ARXIV.2308.00840.
- 11 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.
- 12 Subhash Khot, Dor Minzer, and Muli Safra. On independent sets, 2-to-2 games, and grassmann graphs. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 576–589. ACM, 2017. doi:10.1145/3055399.3055432.
- 13 Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 592–601. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00062.
- 14 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008. URL: <https://doi.org/10.1016/j.jcss.2007.06.019>, doi:10.1016/J.JCSS.2007.06.019.
- 15 Dénes König. Gráfok és mátrixok. *Matematikai és Fizikai Lapok*, 38:116–119, 1931.
- 16 Alexandr Kostochka and Jaroslav Nešetřil. Chromatic number of geometric intersection graphs. In *1995 Prague Midsummer Combinatorial Workshop*, volume 95, pages 43–45. KAM Series, 1995.
- 17 Alexandr V. Kostochka and Jaroslav Nešetřil. Coloring relatives of intervals on the plane, I: chromatic number versus girth. *Eur. J. Comb.*, 19(1):103–110, 1998. URL: <https://doi.org/10.1006/eujc.1997.0151>, doi:10.1006/EUJC.1997.0151.
- 18 Jan Kratochvíl. String graphs. II. recognizing string graphs is np-hard. *J. Comb. Theory B*, 52(1):67–78, 1991. doi:10.1016/0095-8956(91)90091-W.
- 19 Jan Kratochvíl and Jiri Matoušek. String graphs requiring exponential representations. *J. Comb. Theory B*, 53(1):1–4, 1991. doi:10.1016/0095-8956(91)90050-T.
- 20 Andrei A. Krokhin and Jakub Oprsal. An invitation to the promise constraint satisfaction problem. *ACM SIGLOG News*, 9(3):30–59, 2022. doi:10.1145/3559736.3559740.
- 21 James R. Lee. private communication.
- 22 James R. Lee. Separators in region intersection graphs. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPICs*, pages 1:1–1:8. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. URL: <https://doi.org/10.4230/LIPICs.ITCS.2017.1>, doi:10.4230/LIPICs.ITCS.2017.1.
- 23 Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. A 1.9999-approximation algorithm for vertex cover on string graphs. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th International Symposium on Computational Geometry, SoCG 2024, June 11-14, 2024, Athens, Greece*, volume 293 of *LIPICs*, pages 72:1–72:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: <https://doi.org/10.4230/LIPICs.SocG.2024.72>, doi:10.4230/LIPICs.SocG.2024.72.
- 24 Sean McGuinness. Colouring arcwise connected sets in the plane I. *Graphs Comb.*, 16(4):429–439, 2000. doi:10.1007/PL00007228.

- 25 Sean McGuinness. Colouring arcwise connected sets in the plane II. *Graphs Comb.*, 17(1):135–148, 2001. doi:10.1007/PL00007235.
- 26 George L. Nemhauser and Leslie E. Trotter Jr. Properties of vertex packing and independence system polyhedra. *Math. Program.*, 6(1):48–61, 1974. doi:10.1007/BF01580222.
- 27 Arkadiusz Pawlik, Jakub Kozik, Tomasz Krawczyk, Michał Lasoń, Piotr Micek, William T. Trotter, and Bartosz Walczak. Triangle-free intersection graphs of line segments with large chromatic number. *J. Comb. Theory B*, 105:6–10, 2014. URL: <https://doi.org/10.1016/j.jctb.2013.11.001>, doi:10.1016/J.JCTB.2013.11.001.
- 28 Miguel Romero, Marcin Wrochna, and Stanislav Žitný. Pliability and approximating max-csps. *J. ACM*, 70(6):41:1–41:43, 2023. doi:10.1145/3626515.
- 29 Bartosz Walczak. Triangle-free geometric intersection graphs with no large independent sets. *Discret. Comput. Geom.*, 53(1):221–225, 2015. URL: <https://doi.org/10.1007/s00454-014-9645-y>, doi:10.1007/S00454-014-9645-Y.