A 1.9999-Approximation Algorithm for Vertex Cover on String Graphs

Daniel Lokshtanov
University of California, Santa Barbara, CA, USA

Fahad Panolan
University of Leeds, UK

Saket Saurabh
The Institute of Mathematical Sciences, HBNI, Chennai, India

Jie Xue
New York University Shanghai, China

Meirav Zehavi
Ben-Gurion University of the Negev, Beer-Sheva, Israel

Abstract

Vertex Cover is a fundamental optimization problem, and is among Karp's 21 NP-complete problems. The problem aims to compute, for a given graph G, a minimum-size set S of vertices of G such that $G - S$ contains no edge. Vertex Cover admits a simple polynomial-time 2-approximation algorithm, which is the best approximation ratio one can achieve in polynomial time, assuming the Unique Game Conjecture. However, on many restrictive graph classes, it is possible to obtain better-than-2 approximation in polynomial time (or even PTASes) for Vertex Cover. In the club of geometric intersection graphs, examples of such graph classes include unit-disk graphs, disk graphs, pseudo-disk graphs, rectangle graphs, etc.

In this paper, we study Vertex Cover on the broadest class of geometric intersection graphs in the plane, known as *string graphs*, which are intersection graphs of any connected geometric objects in the plane. Our main result is a polynomial-time 1.9999-approximation algorithm for Vertex Cover on string graphs, breaking the natural 2 barrier. Prior to this work, no better-than-2 approximation (in polynomial time) was known even for special cases of string graphs, such as intersection graphs of segments.

Our algorithm is simple, robust (in the sense that it does not require the geometric realization of the input string graph to be given), and also works for the weighted version of Vertex Cover. Due to a connection between approximation for Independent Set and approximation for Vertex Cover observed by Har-Peled, our result can be viewed as a first step towards obtaining constant-approximation algorithms for Independent Set on string graphs.

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

Keywords and phrases vertex cover, geometric intersection graphs, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.72

Funding *Saket Saurabh*: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416), and Swarnajayanti Fellowship (No. DST/SJF/MSA01/2017-18).
Meirav Zehavi: Supported by the ERC grant no. 101039913 (PARAPATH).

Acknowledgements The authors would like to thank the anonymous reviewers of SoCG'24 for their insightful comments, which helped improve the exposition of the paper.

1 Introduction

Vertex Cover is a fundamental optimization problem, and is among Karp's 21 NP-complete problems [19]. In this problem, we are given as input a graph G, and the task is to compute a minimum size set $S \subseteq V(G)$ of vertices of G such that $G - S$ contains no edge. Here, $G - S$ denotes the graph obtained from G by removing all vertices in S.

It is well-known that Vertex Cover admits a simple polynomial-time 2-approximation algorithm, which iteratively adds the two endpoints of an edge to the solution and removes them from the graph. This is already the best approximation one can achieve (in polynomial time) on general graphs, assuming the Unique Game Conjecture [21]. However, on many restricted graph classes, it is possible to beat this lower bound and obtain better approximation algorithms for the problem. For instance, Vertex Cover is known to be polynomial-time solvable on bipartite graphs [18]. Also, it admits better-than-2 approximation on dense graphs [20] or sparse graphs (more precisely, degenerate graphs). The latter includes many well-known graph classes, such as planar graphs, bounded-genus graphs, H-minor-free graphs, bounded-degree graphs, etc. On many of these graph classes, Vertex Cover even admits polynomial-time approximation schemes (PTASes) [3, 5, 8, 10, 7, 13, 14, 16].

Another type of graphs on which one can possibly achieve better-than-2 approximation for Vertex Cover are *geometric intersection graphs*. A geometric intersection graph is defined by a set of geometric objects in the plane (or higher dimensional Euclidean spaces), where the objects correspond to vertices and two vertices are connected by an edge if their corresponding objects intersect. Vertex Cover has been studied on many classes of geometric intersection graphs. Known results (for approximation) include PTASes on intersection graphs of unit disks [10], disks [26] (or more generally, fat objects in a fixed dimension [9]), pseudo-disks [5, 15], and a $(1.5 + \varepsilon)$-approximation on intersection graphs of (axis-parallel) rectangles [4].

However, on some more intricate classes of geometric intersection graphs, it is still unclear whether Vertex Cover admits better-than-2 approximation in polynomial time. The simplest example is intersection graphs of plane segments. More generally, one can consider intersection graphs of arbitrary convex objects, or even arbitrary connected objects in the plane. The latter is known as *string graphs*, which form the most general class of geometric intersection graphs in the plane. In this paper, we study Vertex Cover on string graphs. Our main result is a polynomial-time 1.9999-approximation algorithm for the problem, breaking the natural 2 barrier. In fact, we prove the following more general theorem.

▶ **Theorem 1.** *There is a randomized algorithm that, given a string graph G with a weight function $w : V(G) \to \mathbb{R}_{\geq 0}$, computes in expected polynomial time a vertex cover S of G such that $\sum_{v \in S} w(v) \leq 1.9999 \cdot \mathsf{opt}$, where opt is the minimum total weight of a vertex cover of G.*

Our algorithm of Theorem 1 has the following nice features.

- **Simplicity.** Our algorithm is simple, while it non-trivially combines several well-known techniques in graph algorithms and approximation algorithms, such as linear kernelization for Vertex Cover, local-ratio argument, Klein-Plotkin-Rao decomposition, and the polynomial-time exact algorithm for Vertex Cover on bipartite graphs. The entire algorithm can be presented in about 3 pages (see Section 2).
- **Robustness.** An algorithm that deals with geometric intersection graphs is *robust* if it does not require the geometric realization of the input graph to be given. Our algorithm is robust. In contrast, the PTAS on intersection graphs of fat objects [9] and the $(1.5+\varepsilon)$-approximation algorithm on rectangle intersection graphs [4] are not robust. Note that

robustness is especially important when solving problems on string graphs, because the geometric realization of a string graph can have *exponential* complexity in the size of the graph, by the well-known result of Kratochvíl and Matoušek [23].

- **Generality.** As stated in Theorem 1, our algorithm also applies to Weighted Vertex Cover on string graphs, in which the input graph G is given with a weight function $w : V(G) \to \mathbb{R}_{\geq 0}$ and the goal is to find a vertex cover of G with minimum total weight. In contrast, the PTASes on (pseudo-)disk graphs [5, 15, 26] only work for the unweighted case. Furthermore, our algorithm can be applied to even broader classes of graphs. In fact, our algorithm only uses the properties of string graphs when applying Klein-Plotkin-Rao decomposition [22]. By the work of Lee [24], this decomposition applies to not only string graphs but also any graphs excluding a fixed graph as an induced minor. Thus, we have the following more general result.

▶ **Theorem 2.** *For every fixed graph H, there exists a positive number $c_H > 0$ satisfying the following. There is a randomized algorithm that, given a graph G excluding H as an induced minor with a weight function $w : V(G) \to \mathbb{R}_{\geq 0}$, computes in expected polynomial time a vertex cover S of G such that $\sum_{v \in S} w(v) \leq (2 - c_H) \cdot \mathsf{opt}$, where opt is the minimum total weight of a vertex cover of G.*

Aside from the intrinsic interest in the approximability of Vertex Cover, Theorem 1 can be viewed as a stepping stone towards understanding the approximability of Independent Set on string graphs. Independent Set is the dual problem of Vertex Cover, which aims to find a maximum set of vertices in a graph that are pairwise non-adjacent. In terms of exact algorithms, Independent Set is exactly equivalent to Vertex Cover, while in terms of approximation, they behave rather differently. It was known [17] that Independent Set cannot be approximated within a factor of $n^{1-\varepsilon}$ for any constant $\varepsilon > 0$ in polynomial time, assuming P $\neq$ NP (in contrast to the simple 2-approximation for Vertex Cover). However, it can have constant-approximation algorithms or even PTASes on many restrictive graph classes [3, 5, 7, 13, 14, 16].

The approximabiliy of Independent Set on string graphs is an outstanding open problem that has been attacked from multiple angles since the work of Fox and Pach [11]. It was known that Independent Set is NP-hard even on planar graphs and does not admit an EPTAS even on unit-disk graphs. However, the current hardness results do not rule out the possibility that Independent Set on string graphs admits a PTAS. Nevertheless, the best (polynomial-time) approximation for the problem remains the n^{ε} one of Fox and Pach [11]. Better approximation algorithms have been found for a number of important special cases including PTASes for intersection graphs of (pseudo-)disks [5] and the recent constant-approximation algorithms for intersection graphs of rectangles [12, 27]. If one relaxes the running time to *quasipolynomial*, i.e., $n^{\log^{O(1)} n}$, then one can achieve $1 + \varepsilon$ approximation for intersection graphs of any polygons [1]; such an algorithm is usually called a *QPTAS*. Intersection graphs of polygons form the same class as string graphs. However, the QPTAS of [1] is not robust, and thus its running time depends on the total complexity of all polygons representing the vertices of the input graph. As aforementioned, there are (string) graphs which require realizations with exponential complexity. Therefore, in the worst case, the algorithm of [1] runs in time exponential in the size of the graph, even if the given realization is assumed to have minimum complexity.

Interestingly, based on the linear kernelization for Vertex Cover, Har-Peled [15] established a close relation between Independent Set and Vertex Cover in terms of approximation: a $(1 - c)$-approximation algorithm for (weighted) Independent Set on a graph class with running time $T(n)$ implies a $(1+c)$-approximation algorithm for (weighted) Vertex Cover on the same graph class with running time $T(n)+n^{O(1)}$. In particular, a constant-factor approximation for Independent Set, i.e., $(1-c)$-approximation for some constant $c<1$, implies a better-than-2 approximation for Vertex Cover. Therefore, our result in Theorem 1 can be viewed as a first step towards designing constant-approximation algorithms for Independent Set on string graphs.

Notations. We write $[n]=\{1,\dots,n\}$ for a natural number n. For a graph G, we use $V(G)$ to denote the set of vertices of G. A *path* in G is a sequence $\pi=(v_0,v_1,\dots,v_r)$ of vertices of G such that (v_{i-1},v_i) is an edge in G for all $i\in[r]$, where r is called the *length* of π, denoted by $|\pi|$. A *cycle* in G is a path $(v_0,v_1,\dots,v_r)$ in G where $v_0=v_r$. An *odd cycle* in G refers to a cycle γ in G such that $|\gamma|$ is an odd number. A graph is *bipartite* iff it does not contain any odd cycle. For two vertices $s,t\in V(G)$, the notation $\mathsf{dist}_G(s,t)$ denotes the shortest-path distance between s and t in G, that is, the minimum length of a path $(v_0,v_1,\dots,v_r)$ in G satisfying $v_0=s$ and $v_r=t$.

2 The algorithm

For ease of exposition, we first present our algorithm for unweighted case, i.e., prove Theorem 1 in the special case where the weight function $w:V(G)\to\mathbb{R}_{\geq 0}$ is defined as $w(v)=1$ for all $v\in V(G)$. Generalizing it to the weighted case is straightforward and we defer the discussion to Section 3. For a graph H, we denote by $\mathsf{opt}(H)$ the size of a minimum vertex cover of H.

Step 1: Linear kernelization. In the first step, we apply the well-known linear kernelization for Vertex Cover [28], which reduces the problem to the case where the size of a minimum vertex cover is linear in the size of the graph.

▶ **Theorem 3** ([28]). *One can reduce in polynomial time a Vertex Cover instance $\langle G\rangle$ to another Vertex Cover instance $\langle H\rangle$ where H is an induced subgraph of G and $\mathsf{opt}(H)\geq\frac{|V(H)|}{2}$. Furthermore, the reduction is approximation-preserving: a ρ-approximation solution for $\langle H\rangle$ can be converted in polynomial time to a ρ-approximation solution for $\langle G\rangle$.*

Let G be the input string graph and $n=|V(G)|$. Observe that an induced subgraph of a string graph is still a string graph. Therefore, by Theorem 3, we can now assume without loss of generality that $\mathsf{opt}(G)\geq\frac{n}{2}$. It suffices to solve the problem under this assumption.

Step 2: Removing short odd cycles. In the next step, we make use of the following simple but useful observation, which helps remove all short odd cycles in G.

▶ **Observation 4.** *Let $\ell\geq 1$ be an integer. If γ is a simple cycle in G of length $2\ell-1$, then any vertex cover of G contains at least ℓ vertices on γ.*

Proof. Suppose $\gamma=(v_0,v_1,\dots,v_{2\ell-1})$ where $v_0=v_{2\ell-1}$, and $V_\gamma=\{v_1,\dots,v_{2\ell-1}\}$ is the vertex set of γ. Consider a vertex cover $S\subseteq V(G)$ of G. Let $\phi:V(G)\to\{0,1\}$ be the indicator function for membership in S, i.e., $\phi(v)=1$ if $v\in S$ and $\phi(v)=0$ if $v\notin S$. As S is a vertex cover of G, it covers every edge of γ. Thus, $\phi(v_{i-1})+\phi(v_i)\geq 1$ for each $i\in[2\ell-1]$. It follows that

$$2\cdot|S\cap V_\gamma|=\sum_{i=1}^{2\ell-1}(\phi(v_{i-1})+\phi(v_i))\geq 2\ell-1,$$

which implies $|S\cap V_\gamma|\geq\ell-\frac{1}{2}$. Since $|S\cap V_\gamma|$ is an integer, we have $|S\cap V_\gamma|\geq\ell$. ◀

Let $\ell^+ \geq 1$ be a constant to be determined later. For convenience, we call an odd cycle *short* if its length is at most $2\ell^+ - 1$. We compute a maximal packing Γ of short odd cycles in G, i.e., a maximal set of vertex-disjoint short odd cycles in G. Denote by $V(\Gamma)$ the set of all vertices on the cycles in Γ. Note that Γ can be computed by repeatedly finding a short odd cycle in G and removing the vertices on the cycle from G. Finding a short odd cycle takes $n^{O(\ell^+)}$ time by brute-force, and the time complexity can be improved to $n^{O(1)}$ by exploiting the fixed-parameter tractable algorithm for k-Cycle [2]. After computing Γ, we simply add all vertices in $V(\Gamma)$ to our solution, and remove them from G. This is safe because by Observation 4, strictly more than half of the vertices in each cycle in Γ should be included in the solution and thus including all vertices results in a multiplicative error smaller than 2. We shall discuss this formally when analyzing the algorithm. Let $G' = G - V(\Gamma)$ be the resulting graph.

Step 3: Klein-Plotkin-Rao decomposition. Next, it suffices to compute a vertex cover of G'. Set $n' = |V(G')|$. We shall use a powerful decomposition theorem, which was initially obtained for minor-free graphs by Klein, Plotkin, and Rao [22], and can be extended to string graphs using a result of Lee [24]. It basically states that one can remove a small fraction of vertices from a string graph such that any two vertices in the same connected component of the resulting graph are close to each other in the original graph.

▶ **Theorem 5** ([22, 24]). *There exists a randomized algorithm that, for a given string graph H and an integer $\Delta \geq 1$, computes in expected polynomial time a subset $S \subseteq V(H)$ of $O(|V(H)|/\Delta)$ vertices such that for any two vertices u and v lying in the same connected component of $H - S$, we have $\mathsf{dist}_H(u, v) \leq \Delta$.*

Proof. The theorem is a direct implication of Theorem 4.2 of Lee [24] (see the arxiv version). The statement of that theorem[1] is that for every graph H that excludes a subdivided clique $\dot{K}_h$ as an induced minor and every integer Δ, there exists a distribution on the vertex sets of H, such that if a set S is sampled from this distribution, then (i) every vertex of G has probability at most $O(h^2/\Delta)$ to be included in S, and (ii) with probability at least $1 - o(1)$ it holds that for every two vertices u and v lying in the same connected component of $H - S$, we have $\mathsf{dist}_H(u, v) \leq \Delta$. This theorem holds for string graphs in particular, since string graphs exclude a subdivided K_5 as an induced minor. Even though it is not explicitly stated, the proof of this theorem is in fact algorithmic in the sense that there exists a randomized algorithm that given H and Δ, samples S from the above distribution in polynomial time. Condition (i) guarantees that $\mathbb{E}[|S|] = O(|V(H)|/\Delta)$. Fix a constant $c > 0$ satisfying $\mathbb{E}[|S|] < c \cdot |V(H)|/\Delta$. Markov's inequality together with condition (ii) implies that S satisfies $|S| \leq c \cdot |V(H)|/\Delta$ and the condition required in the theorem with probability at least $\Omega(1)$. Since we can check in polynomial time whether S satisfies the conditions, we can re-sample S until it does. The expected number of re-samplings is $O(1)$. ◀

We apply the above theorem with $H = G'$ and $\Delta = \ell^+ - 1$, and let $S \subseteq V(G')$ be the subset obtained. We then include S in our solution. Note that $|S| = O(n'/\Delta)$ by Theorem 5, i.e., $|S| = O(n'/\ell^+)$. If ℓ^+ is sufficiently large, then the size of S is very small compared to n'. As $\mathsf{opt}(G) \geq \frac{n}{2} \geq \frac{n'}{2}$ by our assumption, $|S|$ is only a small fraction of $\mathsf{opt}(G)$ and hence including S in the solution can only increase the multiplicative error slightly.

[1] The original statement of Lee [24] is in fact more general. The statement we give here comes from applying the statement of Lee to the conformal graph (H, ω) where ω is a function that assigns 1 to every vertex of H, and applying the definition of (α, Δ)-random separators with any constant $R > \frac{1}{2}$.

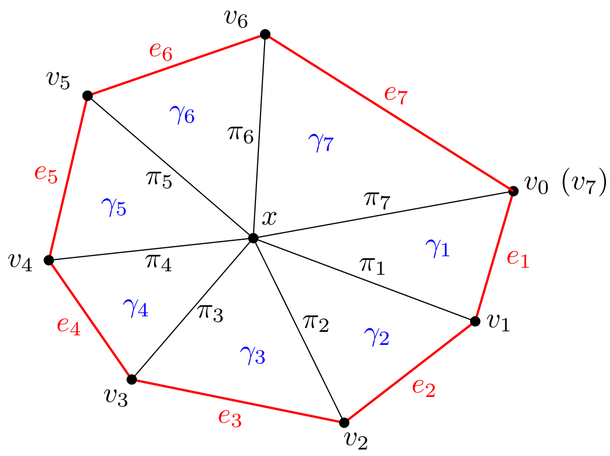

Figure 1 Illustrating the paths $\pi_1, \dots, \pi_r$ and the cycles $\gamma_1, \dots, \gamma_r$.

Step 4: Solving the bipartite case. The remaining task is to compute a (good) vertex cover of $G' - S$. The key observation here is that $G' - S$ is already bipartite!

▶ **Observation 6.** *$G' - S$ is bipartite.*

Proof. It suffices to show that $G' - S$ does not contain odd cycles. Recall that **(i)** G' does not contain odd cycles of length at most $2\ell^+ - 1$ and **(ii)** $\mathsf{dist}_{G'}(u, v) \leq \Delta = \ell^+ - 1$ for any two vertices u and v lying in the same connected component of $G' - S$.

Suppose $G' - S$ contains a cycle $\gamma = (v_0, v_1, \dots, v_r)$ where $v_0 = v_r$ and r is an odd number. For convenience, we write $e_i = (v_{i-1}, v_i)$ for $i \in [r]$. Then $v_1, \dots, v_r$ must lie in the same connected component C of $G' - S$. Pick an arbitrary vertex $x \in C$. We have $\mathsf{dist}_{G'}(x, v_i) \leq \ell^+ - 1$ for all $i \in [r]$, by property **(ii)** above. For each $i \in [r]$, we fix a path π_i in G' from x to v_i of length at most $\ell^+ - 1$. Set $\pi_0 = \pi_r$. Then for each $i \in [r]$, the paths π_{i-1} and π_i together with e_i form a cycle γ_i in G' (which is not necessarily simple). See Figure 1 for an illustration. Observe that the length of each γ_i is at most $2(\ell^+ - 1) + 1 = 2\ell^+ - 1$. We claim that one of the cycles $\gamma_1, \dots, \gamma_r$ is an odd cycle. Indeed,

$$\sum_{i=1}^{r} |\gamma_i| = \sum_{i=1}^{r} (|\pi_{i-1}| + |\pi_i| + 1) = r + \sum_{i=1}^{r} 2 \cdot |\pi_i|,$$

which is an odd number since r is odd. Thus, one of $|\gamma_1|, \dots, |\gamma_i|$ is odd. However, this contradicts property **(i)** above, i.e., G' does not contain odd cycles of at most length $2\ell^+ - 1$. So we conclude that $G' - S$ does not contain odd cycles and is hence bipartite. ◀

Vertex Cover on bipartite graphs is polynomial-time solvable. Therefore, we can compute a minimum vertex cover $S' \subseteq V(G') \backslash S$ of $G' - S$ in polynomial time. Now $V(\Gamma) \cup S \cup S'$ is a vertex cover of G, which is the output of our algorithm.

Putting everything together. We now analyze the approximation ratio of the above algorithm. Let $S^* \subseteq V(G)$ be a minimum vertex cover of G. By assumption, we have $|S^*| \geq \frac{n}{2}$. Recall that $n' = |V(G')|$. As aforementioned, we observe that

$$|S^* \cap V(\Gamma)| \geq \frac{\ell^+}{2\ell^+ - 1} \cdot |V(\Gamma)| = \frac{\ell^+}{2\ell^+ - 1} \cdot (n - n'). \tag{1}$$

To see this, suppose $\mathit{\Gamma} = \{\gamma_1, \ldots, \gamma_r\}$, where $\gamma_1, \ldots, \gamma_r$ are disjoint short odd cycles in G. Suppose $|\gamma_i| = 2\ell_i - 1 \leq 2\ell^+ - 1$ for all $i \in [r]$. By Observation 4, $|S^* \cap V(\gamma_i)| \geq \frac{\ell_i}{2\ell_i - 1} \cdot |\gamma_i| \geq \frac{\ell^+}{2\ell^+ - 1} \cdot |\gamma_i|$. As $\gamma_1, \ldots, \gamma_r$ are disjoint, we have $|S^* \cap V(\mathit{\Gamma})| = \sum_{i=1}^{r} |S^* \cap V(\gamma_i)|$ and $|V(\mathit{\Gamma})| = \sum_{i=1}^{r} |\gamma_i|$. Therefore, Equation 1 holds. Furthermore, note that $|S^* \cap (V(G') \backslash S)|$ is a vertex cover of $G' - S$, which implies $|S^* \cap (V(G') \backslash S)| \geq |S'|$ as S' is a minimum vertex cover of $G' - S$. It follows that

$$|S^* \cap V(G')| \geq |S'|. \tag{2}$$

The last condition we need is $|S| = O(n'/\ell^+)$, by Theorem 5. In fact, here we can be more concrete. A close inspect of the work [24] (proof of Theorem 4.2) reveals that the constant hidden in the $O(\cdot)$ of Theorem 5 can be chosen as any number greater than 2120 (say, 2200). As we applied Theorem 5 with $\Delta = \ell^+ - 1$, we have

$$|S| \leq \frac{2200 \cdot n'}{\ell^+ - 1}. \tag{3}$$

Set $\ell^+ = 4402$. By Equations 1, 2, and 3, we conclude the following.

▶ **Observation 7.** $|V(\mathit{\Gamma}) \cup S \cup S'| - |S^*| \leq 0.49995 \cdot n$. *In particular,* $|V(\mathit{\Gamma}) \cup S \cup S'| \leq 1.9999 \cdot |S^*|$.

Proof. We have $|V(\mathit{\Gamma}) \cup S \cup S'| = |V(\mathit{\Gamma})| + |S| + |S'|$ and $|S^*| = |S^* \cap V(\mathit{\Gamma})| + |S^* \cap V(G')|$. Thus, $|V(\mathit{\Gamma}) \cup S \cup S'| - |S^*| = (|V(\mathit{\Gamma})| - |S^* \cap V(\mathit{\Gamma})|) + (|S| + |S'| - |S^* \cap V(G')|)$. By Equation 1, we have $|V(\mathit{\Gamma})| - |S^* \cap V(\mathit{\Gamma})| \leq \frac{\ell^+ - 1}{2\ell^+ - 1} \cdot (n - n')$. By Equation 2 and 3, we have

$$|S| + |S'| - |S^* \cap V(G')| \leq |S| \leq \frac{2200 \cdot n'}{\ell^+ - 1}.$$

As we set $\ell^+ = 4242$, it then follows that

$$|V(\mathit{\Gamma}) \cup S \cup S'| - |S^*| \leq \frac{\ell^+ - 1}{2\ell^+ - 1} \cdot (n - n') + \frac{2200 \cdot n'}{\ell^+ - 1} \leq \frac{4401}{8803} \cdot n \leq 0.49995 \cdot n.$$

Now recall that $|S^*| \geq \frac{n}{2}$, which implies $|V(\mathit{\Gamma}) \cup S \cup S'| - |S^*| \leq 0.49995 \cdot n \leq 0.9999 \cdot |S^*|$. Finally, we can deduce that $|V(\mathit{\Gamma}) \cup S \cup S'| \leq 1.9999 \cdot |S^*|$. ◀

The above observation shows that our algorithm computes a 1.9999-approximation solution. Finally, we briefly discuss the time complexity of the algorithm. The linear kernelization can be done in $O(n^2 + nm)$ time using the method of Li and Zhu [25] based on crown decomposition. Finding any fixed cycle in a graph can be done in $O(n^{2.376})$ time by the algorithm of Alon et al. [2]. In Step 2, we need to call this algorithm $O(n)$ times, which takes $O(n^{3.376})$ time. This dominates the time cost of Step 3 and 4 (and also the kernelization). Therefore, the overall time complexity of our algorithm is $O(n^{3.376})$. The bound can possibly be improved by designing a faster algorithm for finding the odd cycle packing in Step 2 (instead of repeatedly applying the cycle-finding algorithm). But for simplicity, we are not going to optimize the running time in this paper.

3 Extension to weighted case

Our algorithm directly generalizes to VERTEX COVER with any weight function $w : V(G) \to \mathbb{R}_{\geq 0}$. This is because each of the four steps of the algorithm still works when the vertices have different weights. The linear kernelization (Step 1), Klein-Plotkin-Rao decomposition

(Step 3), and the polynomial-time algorithm for Vertex Cover on bipartite graphs (Step 4) can all be applied to the weighted case. So in what follows, we mainly discuss how to do Step 2 in the weighted setting.

Let $G = (V, E)$ be a string graph with a weight function $w : V(G) \to \mathbb{R}_{\geq 0}$. The kernelization of Theorem 3 works for the weighted case [6]. Applying the result, we can assume that the minimum weight of a vertex cover of G is at least $\frac{1}{2}\sum_{v\in V(G)} w(v)$. To do Step 2, we use the standard local-ratio argument (which extends the argument in the unweighted case).

▶ **Observation 8.** *One can compute in polynomial time a function $\tilde{w} : V(G) \to \mathbb{R}_{\geq 0}$ such that **(i)** for any vertex cover $S \subseteq V(G)$ of G, $\sum_{v\in S} \tilde{w}(v) \geq \frac{\ell^+}{2\ell^+-1} \cdot \sum_{v\in V(G)} \tilde{w}(v)$ and **(ii)** $G[V]$ does not contain odd cycles of length at most $2\ell^+ - 1$, where $V = \{v \in V(G) : w(v) > \tilde{w}(v)\}$.*

Proof. We keep doing the following three steps until G does not contain any short odd cycles (i.e., odd cycles of length at most $2\ell^+ - 1$).

1. Find a short odd cycle γ in G. Set $w_{\mathsf{min}} = \min_{v\in V(\gamma)} w(v)$.
2. $w(v) \leftarrow w(v) - w_{\mathsf{min}}$ for all $v \in V(\gamma)$.
3. Remove from G all vertices $v \in V(G)$ with $w(v) = 0$.

Note that this procedure terminates in at most n iterations, because in each iteration at least one vertex is removed from G (i.e., the vertex on γ with the minimum weight). Suppose it has r iterations. Let γ_i (resp., w_i) denote the short odd cycle γ (resp., the number w_{min}) in the i-th iteration, for $i \in [r]$. Suppose the length of γ_i is $2\ell_i - 1$, where $\ell_i \leq \ell^+$. For each vertex $v \in V(G)$, set $I_v = \{i \in [r] : v \in V(\gamma_i)\}$. We define the function $\tilde{w} : V(G) \to \mathbb{R}_{\geq 0}$ as $\tilde{w}(v) = \sum_{i\in I_v} w_i$. Clearly, a vertex survives in the resulting graph of the above procedure iff $\tilde{w}(v) < w(v)$. In other words, the resulting graph is $G[V]$ for $V = \{v \in V(G) : w(v) > w(v')\}$. Therefore, $G[V]$ does not contain short odd cycles and thus condition (ii) holds. To see condition (i), consider a vertex cover $S \subseteq V(G)$ of G. By Observation 4, $|S \cap V(\gamma_i)| \geq \frac{\ell_i}{2\ell_i-1} \cdot |\gamma_i| \geq \frac{\ell^+}{2\ell^+-1} \cdot |\gamma_i|$, for all $i \in [r]$. Thus,

$$\sum_{v\in S} \tilde{w}(v) = \sum_{v\in S}\sum_{i\in I_v} w_i = \sum_{i=1}^{r} \sum_{v\in S\cap V(\gamma_i)} w_i \geq \sum_{i=1}^{r}\left(\frac{\ell^+}{2\ell^+-1} \cdot w_i|\gamma_i|\right).$$

Note that $\sum_{i=1}^{r} w_i|\gamma_i| = \sum_{v\in V(G)} \tilde{w}(v)$. So we have $\sum_{v\in S} \tilde{w}(v) \geq \frac{\ell^+}{2\ell^+-1} \cdot \sum_{v\in V(G)} \tilde{w}(v)$. ◀

Recall that in Step 2 of the unweighted case, we removed from G the vertices in the short cycles in Γ to obtain the graph G'. Here, we have "fractional" short odd cycles in the proof of Observation 8, and similarly we shall remove them from G to obtain G'. Formally, let $\tilde{w} : V(G) \to \mathbb{R}_{\geq 0}$ be the function in Observation 8. We define $G' = G[V]$ where $V = \{v \in V(G) : w(v) > \tilde{w}(v)\}$ with the new weight function $w' : V(G') \to \mathbb{R}_{\geq 0}$ defined as $w'(v) = w(v) - \tilde{w}(v)$. By Observation 8, G' does not contain any short odd cycles. Step 3 and Step 4 will be applied to the graph G'. For Step 3, the Klein-Plotkin-Rao decomposition (Theorem 5) also works for the weighted case, in which the set S has a small total weight instead of a small size.

▶ **Theorem 9** ([22, 24])**.** *There exists a randomized algorithm that, for a given string graph H with a weight function $w : V(H) \to \mathbb{R}_{\geq 0}$ and an integer $\Delta \geq 1$, computes in expected polynomial time a subset $S \subseteq V(H)$ satisfying $\sum_{v\in S} w(v) = O(\sum_{v\in V(H)} w(v)/\Delta)$ such that for any two vertices u and v lying in the same connected component of $H - S$, $\mathsf{dist}_H(u, v) \leq \Delta$.*

Proof. The algorithm is exactly the same as the one in the proof of Theorem 5. Recall that we can randomly sample a subset $S \subseteq V(H)$ such that (i) every vertex of G has probability $O(1/\Delta)$ to be included in S, and (ii) with probability at least $1 - o(1)$ it holds that for every two vertices u and v lying in the same connected component of $H - S$, we have $\mathsf{dist}_H(u,v) \leq \Delta$. Condition (i) implies that $\mathbb{E}[\sum_{v\in S} w(v)] = O(\sum_{v\in V(H)} w(v)/\Delta)$. Thus, the argument in the proof of Theorem 5 also applies here. ◀

We apply the above theorem on the graph $H = G'$ with the weight function w', and compute the set $S \subseteq V(H)$. Since G' does not contain any short cycles, using the same argument as in the proof of Observation 6, one can show that $G' - S$ is bipartite. It is well-known that the polynomial-time algorithm for Vertex Cover on bipartite graphs also works for the weighted case. Thus, we can compute a minimum-weight vertex cover $S' \subseteq V(G')\backslash S$ of $G' - S$ in polynomial time. As before, our final solution is $U \cup S \cup S'$, where $U = V(G)\backslash V(G_1)$.

The analysis is almost the same as the unweighted case. We give a proof here for completeness. For convenience, we extend the domain of w' to $V(G)$ by setting $w'(v) = 0$ for all $v \in V(G)\backslash V(G_1) = U$. For a subset $V \subseteq V(G)$, we write $W(V) = \sum_{v\in V} w(v)$ and $W'(V) = \sum_{v\in V} w'(v)$. Set $W = W(V(G))$ and $W' = W'(V(G))$. Let $S^* \subseteq V(G)$ be a minimum-weight vertex cover of G. Observation 8 gives an analog of Equation 1, which is

$$W(S^*) - W'(S^*) \geq \frac{\ell^+}{2\ell^+ - 1} \cdot (W - W'). \tag{4}$$

As S' is a minimum-weight vertex cover of $G' - S$, we have an analog of Equation 2, which is

$$W'(S^*) \geq W'(S^* \cap (V(G')\backslash S)) \geq W'(S'). \tag{5}$$

Finally, Theorem 9 gives us the analog of Equation 3, which is

$$W'(S) \leq \frac{2200 \cdot W'}{2\ell^+ - 1}. \tag{6}$$

Now we can apply the same argument as in Observation 7. By the definition of U, we have $W'(U) = 0$ and thus $W(U\cup S\cup S') \leq (W-W')+W'(U\cup S\cup S') = (W-W')+W'(S)+W'(S')$. Furthermore, $W(S^*) = (W(S^*) - W'(S^*)) + W'(S^*)$. Therefore, we have

$$W(U \cup S \cup S') - W(S^*) = \Delta_1 + \Delta_2,$$

where $\Delta_1 = (W - W') - (W(S^*) - W'(S^*))$ and $\Delta_2 = W'(S) + W'(S') - W'(S^*)$. By Equation 4, we have $\Delta_1 \leq \frac{\ell^+-1}{2\ell^+-1} \cdot (W - W')$. Then by Equation 5 and 6, $\Delta_2 \leq \frac{4400\cdot W'}{2\ell^+-1}$. Now set $\ell^+ = 4402$, we directly have $\Delta_1 + \Delta_2 \leq \frac{4401}{8803} \cdot W$ and thus

$$W(U \cup S \cup S') - W(S^*) \leq \frac{4401}{8803} \cdot W \leq 0.49995 \cdot W.$$

Recall that the linear kernelization guarantees $W(S^*) \geq \frac{W}{2}$. Therefore, we finally have $W(U \cup S \cup S') \leq 1.9999 \cdot W(S^*)$. This completes the proof of Theorem 1.

The proof of Theorem 2 is totally the same. Note that our algorithm only uses the properties of a string graph when applying Klein-Plotkin-Rao decomposition (Theorems 5 and 9). Lee [24] showed that these theorems hold for any graphs excluding a fixed graph H as an induced minor, and the constant hidden in $O(\cdot)$ depends on H. Thus, our algorithm also applies to prove Theorem 2.

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.

2 Noga Alon, Raphy Yuster, and Uri Zwick. Color-coding: a new method for finding simple paths, cycles and other small subgraphs within large graphs. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 326–335, 1994.

3 Brenda S. Baker. Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.

4 Reuven Bar-Yehuda, Danny Hermelin, and Dror Rawitz. Minimum vertex cover in rectangle graphs. *Computational Geometry*, 44(6-7):356–364, 2011.

5 Timothy M. Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discret. Comput. Geom.*, 48(2):373–392, 2012.

6 Miroslav Chlebík and Janka Chlebíková. Crown reductions for the minimum weighted vertex cover problem. *Discrete Applied Mathematics*, 156(3):292–312, 2008.

7 Anuj Dawar, Martin Grohe, Stephan Kreutzer, and Nicole Schweikardt. Approximation schemes for first-order definable optimisation problems. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*, pages 411–420. IEEE Computer Society, 2006.

8 Erik D. Demaine and Mohammad Taghi Hajiaghayi. Bidimensionality: new connections between FPT algorithms and ptass. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 590–601. SIAM, 2005.

9 Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM J. Comput.*, 34(6):1302–1323, 2005.

10 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Excluded grid minors and efficient polynomial-time approximation schemes. *J. ACM*, 65(2):10:1–10:44, 2018.

11 Jacob Fox and János Pach. Computing the independence number of intersection graphs. In *Proceedings of the twenty-second annual ACM-SIAM Symposium on Discrete algorithms*, pages 1161–1165. SIAM, 2011.

12 Waldo Gálvez, Arindam Khan, Mathieu Mari, Tobias Mömke, Madhusudhan Reddy Pittu, and Andreas Wiese. A 3-approximation algorithm for maximum independent set of rectangles. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9-12, 2022*, pages 894–905. SIAM, 2022.

13 Martin Grohe. Local tree-width, excluded minors, and approximation algorithms. *Comb.*, 23(4):613–632, 2003.

14 Magnús M. Halldórsson and Jaikumar Radhakrishnan. Greed is good: Approximating independent sets in sparse and bounded-degree graphs. *Algorithmica*, 18(1):145–163, 1997.

15 Sariel Har-Peled. Approximately: Independence implies vertex cover. *arXiv preprint*, 2023. arXiv:2308.00840.

16 Sariel Har-Peled and Kent Quanrud. Approximation algorithms for polynomial-expansion and low-density graphs. *SIAM J. Comput.*, 46(6):1712–1744, 2017.

17 Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

18 John E Hopcroft and Richard M Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.

19 Richard M. Karp. Reducibility among combinatorial problems. In *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*, pages 85–103, 1972.

20 Marek Karpinski and Alexander Zelikovsky. *Approximating dense cases of covering problems.* Citeseer, 1996.

21 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2 - \varepsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

22 Philip Klein, Serge A Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 682–690, 1993.

23 Jan Kratochvíl and Jirí Matousek. String graphs requiring exponential representations. *J. Comb. Theory, Ser. B*, 53(1):1–4, 1991.

24 James R Lee. Separators in region intersection graphs. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. `arXiv:1608.01612`.

25 Wenjun Li and Binhai Zhu. A 2k-kernelization algorithm for vertex cover based on crown decomposition. *Theoretical computer science*, 739:80–85, 2018.

26 Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. A framework for approximation schemes on disk graphs. In *34rd Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2023.

27 Joseph S. B. Mitchell. Approximating maximum independent set for rectangles in the plane. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 339–350. IEEE, 2021.

28 George L Nemhauser and Leslie E Trotter Jr. Vertex packings: structural properties and algorithms. *Mathematical Programming*, 8(1):232–248, 1975.