Genus Characterizes the Complexity of Certain Graph Problems: Some Tight Results*

Jianer Chen[†] Iyad A. Kanj[‡] Ljubomir Perković[‡]
Eric Sedgwick[‡] Ge Xia[§]

Abstract

We study the fixed-parameter tractability, subexponential time computability, and approximability of the well-known NP-hard problems: INDEPENDENT SET, VERTEX COVER, and DOMINATING SET. We derive tight results and show that the computational complexity of these problems, with respect to the above complexity measures, is dependent on the genus of the underlying graph. For instance, we show that, under the widely-believed complexity assumption $W[1] \neq \text{FPT}$, INDEPENDENT SET on graphs of genus bounded by $g_1(n)$ is fixed parameter tractable *if and only if* $g_1(n) = o(n^2)$, and DOMINATING SET on graphs of genus bounded by $g_2(n)$ is fixed parameter tractable *if and only if* $g_2(n) = n^{o(1)}$. Under the assumption that not all SNP problems are solvable in subexponential time, we show that the above three problems on graphs of genus bounded by $g_3(n)$ are solvable in subexponential time *if and only if* $g_3(n) = o(n)$. We also show that the INDEPENDENT SET, the kernelized VERTEX COVER, and the kernelized DOMINATING SET problems on graphs of genus bounded by $g_4(n)$ have PTAS if $g_4(n) = o(n/\log n)$, and that, under the assumption $\text{P} \neq \text{NP}$, the INDEPENDENT SET problem on graphs of genus bounded by $g_5(n)$ has no PTAS if $g_5(n) = \Omega(n)$, and the VERTEX COVER and DOMINATING SET problems on graphs of genus bounded by $g_6(n)$ have no PTAS if $g_6(n) = n^{\Omega(1)}$.

1 Introduction

NP-completeness theory [24] serves as a foundation for the study of intractable computational problems. However, this theory does not obviate the need for solving these hard problems because of their practical importance. Many approaches have been proposed to solve these problems, including polynomial time approximation, fixed parameter tractable computation, and subexponential time algorithms. The INDEPENDENT SET, VERTEX COVER, and DOMINATING SET problems are among the celebrated examples of such problems. Unfortunately, these problems refuse to give in to most of these approaches. It is known [5] that none of them has a polynomial time approximation scheme unless P = NP. It is also unlikely that any of them is solvable in subexponential time [27]. In terms of fixed parameter tractability, INDEPENDENT SET and DOMINATING SET do not seem to have efficient algorithms even for small parameter values [19].

*A preliminary version of this work was reported at *The Thirtieth International Colloquium on Automata, Languages, and Programming* (ICALP 2003), Eindhoven, The Netherlands, June 30–July 4, 2003, and published in *Lecture Notes in Computer Science 2719*, pp. 845-856, (2003). The research is supported in part by the NSF under the Grants CCR-0311590 and CCF-0430683, and by DePaul University Competitive Research Grant.

[†]Corresponding author. Department of Computer Science, Texas A&M University, College Station, TX 77843-3112. Email: chen@cs.tamu.edu.

[‡]School of Computer Science, Telecommunications, and Information Systems, DePaul University, 243 S. Wabash Avenue, Chicago, IL 60604-2301. Email: {ikanj,lperkovic,esedgwick}@cs.depaul.edu.

[§]Department of Computer Science, Texas A&M University, College Station, TX 77843-3112. Email: gexia@cs.tamu.edu.

Prob.	FPT		Subexp. Time		Approximability	
	Our Results	Previous	Our Results	Previous	Our Results	Previous
VC	–	FPT [19]	$2^{o(n)}$ iff $g=o(n)$	$2^{O(\sqrt{n})}$ if $g=c$ [3, 30]	PTAS¶ if $g=o(\frac{n}{\log n})$ APX-C if $g=n^{\Omega(1)}$	PTAS if $g=c$ [7, 30]
IS	FPT iff $g=o(n^2)$	FPT if $g=0$ [3]	$2^{o(n)}$ iff $g=o(n)$	$2^{O(\sqrt{n})}$ if $g=c$ [3, 30]	PTAS if $g=o(\frac{n}{\log n})$ APX-H if $g=\Omega(n)$	PTAS if $g=c$ [7, 30]
DS	FPT iff $g=n^{o(1)}$	FPT if $g=c$ [20]	$2^{o(n)}$ iff $g=o(n)$	$2^{O(\sqrt{n})}$ if $g=c$ [3, 30]	PTAS¶ if $g=o(\frac{n}{\log n})$ APX-H if $g=n^{\Omega(1)}$	PTAS if $g=c$ [7, 30]

¶Only true for kernelized graphs, see Theorem 4.5 and Theorem 4.6.

Table 1: Comparison between our results and the previous results

Variants of these problems were studied as well where the input graph is constrained to have certain structural properties (e.g., bounded degree graphs and planar graphs) [1, 3, 6, 20, 24]. In particular, the problems on the class of planar graphs (the problems remain NP-hard) become more tractable in terms of the above three complexity measures. All of the three problems on planar graphs have polynomial time approximation schemes [7, 30], and are solvable in subexponential time [30]. Recent research in fixed parameter tractability shows that all the three problems admit parameterized algorithms whose running time is subexponential in the parameter [3]. This line of research has attracted considerable recent interests and the results have been extended to graphs of bounded genus [16, 20, 23].

This raises an interesting question: What are the graph structures that determine the computational complexity of these important NP-hard problems?

In this paper, we demonstrate how the genus of the underlying graph plays an important role in characterizing the parameterized complexity, the subexponential time computability, and the approximability of the VERTEX COVER, INDEPENDENT SET, and DOMINATING SET problems. Our research shows that in most cases, graph genus is the sole factor that determines the complexity of the above problems. More precisely, in most cases, there is a precise genus threshold that determines the computational complexity of the problems in terms of the three complexity measures. For instance, we show that under the widely-believed complexity assumption $W[2] \neq$ FPT, DOMINATING SET is fixed parameter tractable *if and only if* the graph genus is $n^{o(1)}$. This result significantly extends both Alber *et al.* and Ellis *et al.*'s results for planar graphs and for constant genus graphs [1, 20]. The proof is also simpler and more uniform. It is also shown that under the assumption $W[1] \neq$ FPT, INDEPENDENT SET is fixed parameter tractable *if and only if* the graph genus is $o(n^2)$. For the subexponential time computability, we show that under the assumption that not all SNP problems are solvable in subexponential time, VERTEX COVER, INDEPENDENT SET, and DOMINATING SET are solvable in subexponential time *if and only if* the genus of the graph is $o(n)$. In terms of approximability, we show that graph genus has a direct impact on whether INDEPENDENT SET, VERTEX COVER, and DOMINATING SET have polynomial time approximation schemes. A summary of our main results and the previous known results is given in Table 1.

We make two remarks on our results. First, all our tractability results are robust [21] in the sense that our algorithms work correctly regardless of whether the input graphs satisfy the required genus bound $g(n)$. As long as the input graphs satisfy the required genus bound $g(n)$, our algorithms construct correct solutions for the problems; whereas when our algorithms fail in constructing a solution, they correctly report that the genus of the input graph exceeds the required bound $g(n)$. Second, the techniques proposed in the current paper are not restricted to only the above three problems, and can be extended to derive similar results for other NP-hard graph problems.

We give a quick review on the related terminologies. Let G be a simple and undirected graph. A set of vertices C is a *vertex cover* for G if every edge in G is incident to at least one vertex in C. An *independent set* I in G is a subset of vertices such that no two vertices in I are adjacent. A *dominating set* D in G is a set of vertices such that every vertex in G is either in D or adjacent to a vertex in D. The VERTEX COVER (resp. INDEPENDENT SET, DOMINATING SET) problem is for a given graph G to construct a vertex cover of minimum size (resp. an independent set of maximum size, a dominating set of minimum size).

A *surface* of genus g is a sphere with g handles in the 3-space [25]. A graph G *embedded* in a surface S is a continuous one-to-one mapping from the graph into the surface. The embedding is *cellular* if each component of $S - G$, which is called a *face*, is homeomorphic to an open disk [25]. In this paper, we only consider cellular graph embeddings. The *size* of a face is the number of edge sides along the boundary of the face. The (*minimum*) *genus* $\gamma_{\min}(G)$ of a graph G is the smallest integer g such that G has an embedding on a surface of genus g. For more detailed discussions on data structures and algorithms for graph embedding on surfaces, the readers are referred to [9].

2 Genus and parameterized complexity

Parameterized complexity theory [19] was motivated by the observation that many important NP-hard problems in practice are associated with a parameter whose value usually falls within a small or a moderate range. Thus, taking the advantage of the small size of the parameter may significantly speedup the computation. We briefly review the basic concepts and refer the readers to [19] for more details.

A *parameterized problem* consists of instances of the form (x, k), where x is the *problem description* and k is an integer called the *parameter*. For instance, the VERTEX COVER problem can be parameterized so that each instance of it is of the form (G, k), where G is a graph and k is the parameter, asking whether the graph G has a vertex cover of k vertices. Similarly, we can define the parameterized versions for INDEPENDENT SET and DOMINATING SET. A parameterized problem Q is *fixed parameter tractable* if it can be solved by an algorithm of running time $f(k)n^c$, where f is a function independent of $n = |x|$ and c is a constant. Denote by FPT the class of all fixed parameter tractable problems. An example of the FPT problems is the VERTEX COVER problem that can be solved in time $O(1.285^k + kn)$ [13]. On the other hand, a large class of computational problems seems not to belong to the class FPT. A hierarchy of parameterized intractability, the W-hierarchy, has been introduced [19]. The 0th level of the hierarchy is the class FPT, and the ith level is denoted by $W[i]$ for $i > 0$. A parameterized complexity preserving reduction (the *fpt-reduction*) has been defined as follows: a parameterized problem Q is *fpt-reducible* to another parameterized problem Q' if there is an algorithm of running time $f(k)|x|^c$ that on an instance (x, k) of Q produces an instance $(x', g(k))$ of Q', such that (x, k) is a yes-instance of Q if and only if $(x', g(k))$ is a yes-instance of Q', where the functions $f(k)$ and $g(k)$ depend only on k, and c is a constant. A parameterized problem Q is $W[i]$-*hard* if every problem in $W[i]$ is fpt-reducible to Q, and is $W[i]$-*complete* if in addition Q is in $W[i]$. In particular, if any $W[i]$-hard problem is in FPT, then $W[i] =$ FPT, which, to the common belief, is very unlikely.

2.1 Genus and INDEPENDENT SET

The parameterized INDEPENDENT SET problem (or simply INDEPENDENT SET without any confusion) is a representative of the $W[1]$-complete problems [19]. Thus, it is unlikely to be fixed parameter tractable. Actually, very recent research has shown strong evidence that it is even unlikely that the problem is solvable in time $n^{o(k)}$ [10, 11]. In this subsection, we discuss how graph genus affects the parameterized complexity of INDEPENDENT SET.

Theorem 2.1 *The* INDEPENDENT SET *problem on graphs of genus bounded by $g(n)$ is fixed parameter tractable if $g(n) = o(n^2)$.*

PROOF. Since $g(n) = o(n^2)$, there is a nondecreasing and unbounded function $r(n)$ such that $g(n) \leq n^2/r(n)$.[1] Without loss of generality, we can assume that $r(n) \leq n^2$. Otherwise, $g(n) = 0$, and the theorem follows from [3]. Let G be a graph of n vertices and genus $g' \leq g(n)$. Recall that the *chromatic number* $\chi(G)$ of G is the smallest integer p such that G can be colored with p colors so that no two adjacent vertices are colored with the same color. By Heawood's Theorem [25], the chromatic number $\chi(G)$ of the graph G is bounded by $(7 + \sqrt{1 + 48g'})/2$. From the definition, the chromatic number $\chi(G)$ of G implies an independent set of at least $n/\chi(G)$ vertices in G. Thus, the size $\alpha(G)$ of a maximum independent set in the graph G is at least $2n/(7 + \sqrt{1 + 48g'})$. Since $g' \leq g(n) \leq n^2/r(n)$, we get (note that $r(n) \leq n^2$)

$$\alpha(G) \geq \frac{2n}{7 + \sqrt{1 + 48n^2/r(n)}} = \frac{2n\sqrt{r(n)}}{7\sqrt{r(n)} + \sqrt{r(n) + 48n^2}} \geq \frac{2n\sqrt{r(n)}}{7n + \sqrt{n^2 + 48n^2}} = \frac{\sqrt{r(n)}}{7} \qquad (1)$$

Now we are ready for describing our parameterized algorithm. Note that one difficulty we must overcome is estimating the genus of the input graph. The graph minimum genus problem is NP-complete [32], and there is no known effective approximation algorithm for the problem. Therefore, some special tricks have to be used for this purpose. Here we will make use of the approximation algorithm for the graph minimum genus problem proposed in [12], which on an input graph G constructs an embedding of G whose genus is bounded by $\max\{4\gamma_{\min}(G), \gamma_{\min}(G) + 4n\}$. Consider the algorithm given in Figure 1.

ALGORITHM. IS-FPT
Input: a graph G of n vertices and an integer k
Output: decide if G has an independent set of k vertices

1. let $r_1(n) = \min\{r(n)/4, nr(n)/(n + 4r(n))\}$;
2. construct an embedding $\pi(G)$ of G using the algorithm in [12];
3. **if** the genus of $\pi(G)$ is larger than $n^2/r_1(n)$ **then** Stop ("the genus of G is larger than $g(n)$");
4. **if** $k \leq \sqrt{r_1(n)}/7$ **then** Stop ("the graph G has an independent set of k vertices")
   else try all vertex subsets of k vertices to derive a conclusion.

Figure 1: A parameterized algorithm for INDEPENDENT SET

We analyze the time complexity of the algorithm **IS-FPT**. First note that by our assumption on the function $r(n)$, the function $r_1(n)$ is also nondecreasing and unbounded. The embedding $\pi(G)$ of the graph G in step 2 can be constructed in linear time [12], and the genus of the embedding $\pi(G)$ can also be computed in linear time [9].

Since $r_1(n) = \min\{r(n)/4, nr(n)/(n + 4r(n))\}$, if the genus $\gamma(\pi(G))$ of the embedding $\pi(G)$ is larger than $n^2/r_1(n)$, then $\gamma(\pi(G))$ is larger than both $4n^2/r(n)$ and $n^2/r(n) + 4n$. According to [12], the genus $\gamma(\pi(G))$ of the embedding $\pi(G)$ is bounded by $\max\{4\gamma_{\min}(G), \gamma_{\min}(G) + 4n\}$. Thus, in case $\gamma(\pi(G)) \leq 4\gamma_{\min}(G)$, we have $4\gamma_{\min}(G) > 4n^2/r(n)$, and in case $\gamma(\pi(G)) \leq \gamma_{\min}(G) + 4n$,

[1] In this paper, we only consider "simple" complexity functions whose value can be feasibly computed. Thus, in our discussion, the computational time for computing the values of complexity functions as such $g(n)$ and $r(n)$ will be neglected.

we have $\gamma_{\min}(G) + 4n > n^2/r(n) + 4n$. Thus, in all cases, we will have $\gamma_{\min}(G) > n^2/r(n) \geq g(n)$. In consequence, the algorithm **IS-FPT** concludes correctly if it stops in step 3.

If the algorithm **IS-FPT** reaches step 4, we know that the minimum genus of the graph G is bounded by $n^2/r_1(n)$. By the above analysis and the relation in (1), the size of a maximum independent set in G is at least $\sqrt{r_1(n)}/7$. Thus, in case $k \leq \sqrt{r_1(n)}/7$, there must be an independent set in G with k vertices. On the other hand, if $k > \sqrt{r_1(n)}/7$ then $\overline{r_1}(49k^2) \geq n$, where $\overline{r_1}$ is the inverse function of the function $r_1(n)$ defined by $\overline{r_1}(p) = \min\{\ q \mid r_1(q) \geq p\ \}$. Since the function $r_1(n)$ is nondecreasing and unbounded, it is not difficult to see that the inverse function $\overline{r_1}(p)$ is also nondecreasing and unbounded. Since enumerating all vertex subsets of k vertices in the graph G can be done in $O(2^n)$ time, which is bounded by $O(2^{\overline{r_1}(49k^2)})$, we conclude that the total running time of the algorithm **IS-FPT** is bounded by $O(f(k) + n^2)$, where $f(k) = 2^{\overline{r_1}(49k^2)}$ is a function dependent only on k and not on n.

Thus, the algorithm **IS-FPT** solves the INDEPENDENT SET problem on graphs of genus bounded by $g(n)$ in time $O(f(k) + n^2)$, and the problem is fixed parameter tractable. □

Remark. The algorithm **IS-FPT** does not have to know whether the input graph has its minimum genus bounded by $g(n)$. Moreover, the algorithm **IS-FPT** does not need to decide precisely whether the input graph has a minimum genus bounded by $g(n)$. In fact, on some graphs whose minimum genus is larger than $g(n)$, the algorithm **IS-FPT** may still be able to decide correctly whether the graphs have an independent set of size k. The point is, if the input graph has its minimum genus bounded by $g(n)$, then the algorithm **IS-FPT**, without needing to know this fact, will definitely and correctly decide whether it has an independent set of size k.

Theorem 2.2 *The* INDEPENDENT SET *problem on graphs of genus bounded by $g(n)$ is $W[1]$-complete if $g(n) = \Omega(n^2)$.*

PROOF. Since INDEPENDENT SET on general graphs is $W[1]$-complete [19], it suffices to show that INDEPENDENT SET on general graphs is fpt-reducible to INDEPENDENT SET on graphs of genus bounded by $g(n)$. Since $g(n) = \Omega(n^2)$, we assume $g(n) \geq cn^2$, where c is a constant.

Let G_1 be an arbitrary graph with n_1 vertices. It is well-known that the genus g_1 of G_1 is always bounded by $(n_1 - 3)(n_1 - 4)/12 \leq n_1^2/12$ [25]. Thus, if $c \geq 1/12$ then G_1 already has its genus bounded by cn_1^2. Otherwise, we construct a new graph G_2 as follows. G_2 contains $h = \lceil 1/(12c) \rceil > 1$ copies of the graph G_1. Partition the h copies of G_1 arbitrarily into two nonempty groups $\mathcal{A}_1$ and $\mathcal{A}_2$, and pick any pair of adjacent vertices u_1 and v_1 in G_1. Now introduce a new edge $[u_2, v_2]$, where u_2 and v_2 are two new vertices. Connect u_2 to the vertex u_1 in each copy of G_1 in the group $\mathcal{A}_1$ and connect v_2 to the vertex v_1 in each copy of G_1 in the group $\mathcal{A}_2$. This completes the construction of the graph G_2. It is not difficult to verify that the graph G_1 has an independent set of k_1 vertices if and only if the graph G_2 has an independent set of $k_2 = hk + 1$ vertices. Thus, the reduction from (G_1, k_1) to (G_2, k_2) is an fpt-reduction. Moreover, the graph G_2 has $n_2 = hn_1 + 2$ vertices and we can verify [25] that the genus of G_2 is $g_2 = hg_1$. Thus, we have

$$g_2 = hg_1 \leq \frac{hn_1^2}{12} = \frac{(hn_1)^2}{12h} \leq \frac{n_2^2}{12/(12c)} = cn_2^2 \leq g(n_2)$$

Thus, the genus of the graph G_2 of n_2 vertices is bounded by $g(n_2)$.

This completes the fpt-reduction that reduces an instance (G_1, k_1) of INDEPENDENT SET on general graphs to an instance (G_2, k_2) of INDEPENDENT SET on graphs of genus bounded by $g(n)$. In consequence, INDEPENDENT SET on graphs of genus bounded by $g(n)$ is $W[1]$-complete. □

Combining Theorem 2.1 and Theorem 2.2, and noting that the genus of a graph of n vertices is always bounded by $(n-3)(n-4)/12$ [25], we have the following tight result.

Corollary 2.3 *Assuming $FPT \neq W[1]$, the* INDEPENDENT SET *problem on graphs of genus bounded by $g(n)$ is not fixed parameter tractable if and only if $g(n) = \Theta(n^2)$.*

2.2 Genus and DOMINATING SET

DOMINATING SET is the most well-known $W[2]$-complete problem [19]. Thus, it is even "harder" than INDEPENDENT SET in terms of its parameterized complexity. Recently, there has been considerable interest in developing parameterized algorithms for DOMINATING SET on graphs of small genus [1, 3, 16, 17, 20, 22, 23, 29]. In particular, it is known that DOMINATING SET on planar graphs [1, 3] and on graphs of constant genus [16, 17, 20, 23] is fixed parameter tractable. We will show a much stronger result in this subsection: DOMINATING SET on graphs of genus bounded by $g(n)$ is fixed-parameter tractable if and only if $g(n) = n^{o(1)}$.

For a given instance (G, k) of DOMINATING SET, we apply a branch-and-bound process to construct a dominating set D of k vertices in G. Initially, $D = \emptyset$. In a more general form during the process, suppose we have correctly included certain vertices in the dominating set D, and removed these vertices from the graph G. The vertices in the remaining graph G' are colored either "white" or "black", where each white vertex is adjacent to a vertex in D (thus needs no further domination) and each black vertex is adjacent to no vertex in D (thus still needs to be dominated in the remaining graph G'). The graph G' thus will be called a *BW-graph*. We call a set D' of vertices in the BW-graph G' a *B-dominating set* if every black vertex in G' is either in D' or is adjacent to a vertex in D'. Note that if the current set D has d vertices, then the graph G has a dominating set of k vertices, including all vertices in D, if and only if the BW-graph G' has a B-dominating set of $k - d$ vertices. Thus, our task is to construct a B-dominating set of $k - d$ vertices in the BW-graph G'.

Certain reduction rules can be applied to a BW-graph G':

R1. Remove from G' all edges between white vertices;
R2. Remove from G' all white vertices of degree 1;
R3. If all neighbors of a white vertex u_1 are neighbors of another white vertex u_2, remove u_1 from G'.

Let G'' be a BW-graph after applying any of the above rules on G'. It is known [1, 20] that there is a B-dominating set of k vertices in G' if and only if there is a B-dominating set of k vertices in G''. A BW-graph G is called *reduced* if none of the above rules can be applied. According to rule **R1**, every edge in a reduced BW-graph either connects two black vertices or connects a black vertex and a white vertex (the edge will be called a *bb-edge* or a *bw-edge*, respectively).

We will show that in a reduced BW-graph, the number of black vertices will not be very small. For this purpose, we first need to give a brief discussion on certain basic facts about graph embeddings. For more detailed and formal proofs of these facts, the readers are referred to [9].

Fact 1. A face of size 1 can only be made by a self-loop, and a face of size 2 must be made by two multiple edges on the same pair of vertices.

Fact 2. Let F be a face of size d in a graph embedding with boundary vertices $u_1, u_2, \ldots, u_d$, cyclically ordered along the face boundary. If we run a new edge from u_1 to u_i crossing the face F, $1 \leq i \leq d$, then the face F is split into two faces of sizes i and $d - i + 2$, respectively, both having the new edge on their face boundaries. No other faces in the embedding are changed. Moreover, the embedding genus is unchanged.

Fact 3. In a given embedding of a graph G, the neighbors of every vertex u in G specify a unique cyclic order $[u_1, u_2, \ldots, u_d]$ so that the edges $[u, u_1]$, $[u, u_2]$, $\ldots$, $[u, u_d]$ form a cyclic order around the vertex u in a small region on the embedding. In particular, if every triple (u, u_i, u_{i+1}), $i = 1, 2, \ldots, d$ (here we take u_{d+1} as u_1), makes a triangle face on the embedding, then removing the vertex u (and all edges incident on u) will merge all these triangle faces into a single face of size d. The embedding genus and all other faces are unchanged.

Fact 4. Suppose there is a triangle face (u_1, u_2, u_3) in an embedding, the vertex u_1 has degree 2, and there are no multiple edges between u_2 and u_3, then removing the vertex u_1 and the two edges incident on u_1 neither changes the embedding genus nor creates a face of size less than 3.

The following lemma can be easily derived from the famous Euler Polyhedral Equation [25].

Lemma 2.4 *If G is a graph of n vertices and m edges (with possibly multiple edges and self-loops), and G has an embedding on a surface of genus g such that all faces of the embedding have size at least 3, then $m \leq 6g + 3n - 6$.*

Now we are ready to prove the following important lemma, which derives relations among the numbers of black vertices, white vertices, edges, and the genus of a reduced BW-graph.

Lemma 2.5 *Let G be a reduced BW-graph of minimum genus g, with m edges and n vertices, in which n_w are white and n_b are black, and suppose that G has neither multiple edges nor self-loops, then* (a) $m \leq 9n_b + 18g - 18$; *and* (b) $n \leq 4n_b + 6g - 6$.

Proof. Let $\pi(G)$ be an embedding of genus g for the graph G. By rules **R1** and **R2**, the degree of a white vertex u in G is at least 2 and all neighbors of u are black. We perform the following operations on each white vertex u.

If the white vertex u has degree 2 with two black neighbors u_1 and u_2, and there is no edge between u_1 and u_2, then we add a new edge $[u_1, u_2]$ crossing a face in the embedding to make a triangle face with u (note that since u has degree 2, this is always possible). Adding the new edge $[u_1, u_2]$ does not create a face of size less than 3, because it does not introduce new self-loops or new multiple edges. Moreover, the embedding genus is unchanged.

If u has degree $d > 2$ and u_1, u_2, $\ldots$, u_d are the d black neighbors of u, ordered in clockwise order around u in the embedding, then for each pair of vertices u_i and u_{i+1}, $i = 1, 2, \ldots, d$ (here we take u_{d+1} as u_1), if the vertices u, u_i, u_{i+1} do not form a triangle face in the embedding $\pi(G)$, then we add a new edge $[u_i, u_{i+1}]$, crossing a face in the embedding $\pi(G)$, to make a triangle face (u, u_i, u_{i+1}) (again, this is always possible). This does not change the embedding genus. Note that adding this new edge may create multiple edges between u_i and u_{i+1}. However, the new edge does not create any faces of size less than 3. This can be proved as follows. First this does not create faces of size 1 because it does not create self-loops. Second, if it created a face of size 2, then the two sides of the new edge $[u_i, u_{i+1}]$ are on the face boundaries of a face of size 2 and a face of size 3 (i.e., the triangle face (u, u_i, u_{i+1})). This, according to Fact 2, would imply that before adding the new edge $[u_i, u_{i+1}]$, the vertices u, u_i, u_{i+1} had already made a triangle face. This proves that adding the new edge $[u_i, u_{i+1}]$ does not create faces of size less than 3. Finally, note that the vertices u_i and u_{i+1} cannot be the neighbors of a white vertex of degree 2 – otherwise by rule **R3**, the white vertex of degree 2 would have been removed. Thus, processing white vertices of degree larger than 2 does not create multiple edges for white vertices of degree 2.

Since the graph G has neither self-loops nor multiple edges, by Fact 1, the embedding $\pi(G)$ has all its faces of size at least 3. Let G' be the graph and $\pi(G')$ be the embedding of G' after applying the above process on all white vertices in G. By the above discussion, the embedding $\pi(G')$ has genus g and all faces in $\pi(G')$ have size at least 3. We estimate the number m_{bb} of bb-edges in the graph G'. For each white vertex u of degree 2 with neighbors u_1 and u_2 in G', we associate the bb-edge $[u_1, u_2]$ with the two bw-edges $[u, u_1]$ and $[u, u_2]$. For each white vertex u of degree $d > 2$ with neighbors u_1, u_2, ..., u_d in G', for each $i = 1, 2, \ldots, d$ (here we take $u_{d+1} = u_1$), we associate the bb-edge $[u_i, u_{i+1}]$ that is on the boundary of the triangle face (u, u_i, u_{i+1}) with the bw-edge $[u, u_i]$. Note that each such bb-edge $[u_i, u_{i+1}]$ can be associated with at most two bw-edges because each edge can be on the boundaries of at most two faces. Moreover, the bb-edge $[u_i, u_{i+1}]$ cannot be associated with the two bw-edges incident on any degree-2 white vertex since u_i and u_{i+1} cannot be the neighbors of a degree-2 white vertex in G' (see the discussion in the last paragraph). Since every bw-edge must be incident on a white vertex, the above association shows that the number m_{bw} of bw-edges is at most twice of the number m_{bb} of bb-edges in G': $m_{bw} \leq 2m_{bb}$. Since the bw-edges in the graph G' are the same as those in the graph G, and the number of bb-edges in G is no more than that in G', we obtain

$$m \leq m_{bw} + m_{bb} \leq 3m_{bb} \tag{2}$$

Moreover, since each white vertex in G has degree at least 2, it is easy to see that the number n_w of white vertices in G is at most half the number m_{bw} of bw-edges in G. Thus,

$$n_w \leq m_{bw}/2 \leq m_{bb} \tag{3}$$

Recall that the embedding $\pi(G')$ has genus g and all faces in $\pi(G')$ have size at least 3. Now we remove all white vertices from the graph G' and from the embedding $\pi(G')$. Let the resulting graph and embedding be G'' and $\pi(G'')$, respectively. By Fact 3 and Fact 4, removing a white vertex neither changes the embedding genus nor creates faces of size less than 3. Thus, the embedding $\pi(G'')$ has genus g and all faces in $\pi(G'')$ have size at least 3. Note that the number of edges in G'' is equal to the number m_{bb} of bb-edges in G', and the number of vertices in G'' is equal to the number n_b of black vertices in G. Applying Lemma 2.4 to the graph G'', we get

$$m_{bb} \leq 6g + 3n_b - 6$$

Replacing m_{bb} by $6g + 3n_b - 6$ in relations (2) and (3), and noting that $n = n_w + n_b$ complete the proof of the lemma. □

Now we are ready to prove the following theorem.

Theorem 2.6 *The* DOMINATING SET *problem on graphs of genus bounded by $g(n)$ is fixed parameter tractable if $g(n) = n^{o(1)}$.*

PROOF. Since $g(n) = n^{o(1)}$, we can write $g(n) \leq n^{1/r(n)}$ for some nondecreasing and unbounded function $r(n)$. For an instance (G, k) of the DOMINATING SET problem, where the graph G has n vertices and genus g', we apply the algorithm **DS-FPT** in Figure 2.

Let $\bar{r}$ be the inverse function of the function $r(n)$ defined by $\bar{r}(p) = \min\{ q \mid r(q) \geq p \}$. Then the function $\bar{r}$ is also nondecreasing and unbounded. In case $k \geq r(n)$, we have $\bar{r}(k) \geq n$. Thus, step 1 of the algorithm DS-FPT takes time $O(2^n) = O(2^{\bar{r}(k)})$.

Now suppose $k < r(n)$, step 3 repeatedly branches at a black vertex of degree bounded by 19 in the reduced BW-graph G_0. The search tree size $T(k)$ of step 3 thus satisfies the recurrence relation

$$T(k) \leq 20 \cdot T(k-1)$$

ALGORITHM. DS-FPT
Input: a graph G of n vertices and an integer k
Output: decide if G has a dominating set of k vertices

1. **if** $k \geq r(n)$ **then** solve the problem by enumerating all subsets of k vertices in G; Stop;
2. $k_0 = k$; $D = \emptyset$; $G_0 = G$; color all vertices of G_0 black;
3. **while** there is a black vertex u of degree $d \leq 19$ in G_0 **do**
3.1. make a $(d+1)$-way branch, each includes either u or a neighbor of u in D;
3.2. remove the new vertex in D from G_0, and color its neighbors in G_0 white;
3.3. apply rules R1-R3 to make G_0 a reduced BW-graph;
3.4. $k_0 = k_0 - 1$;
4. **if** the graph G_0 has at most $78n^{1/k}$ vertices
4.1. **then** find a B-dominating set of k_0 vertices in G_0 by enumerating all vertex subsets of k_0 vertices in G_0
4.2. **else** Stop ("the graph G has genus larger than $g(n)$");

Figure 2: A parameterized algorithm for DOMINATING SET

which has a solution $T(k) = O(20^k)$.

At the end of step 3, all black vertices in the reduced BW-graph G_0 have degree at least 20. Suppose at this point, the number of edges, the number of vertices, and the number of black vertices in G_0 are m_0, n_0 and n_b, respectively. Since $2m_0$ is equal to the sum of total vertex degrees in G_0, we have $2m_0 \geq 20n_b$. By Lemma 2.5(a), we also have $m_0 \leq 9n_b + 18g' - 18$ (note that the genus of the reduced BW-graph G_0 cannot be larger than the genus g' of the original graph G). Combining these two relations, we get $n_b \leq 18g' - 18$. By Lemma 2.5(b), we have $n_0 \leq 4n_b + 6g' - 6$. Thus

$$n_0 \leq 4n_b + 6g' - 6 \leq 78g' - 78 < 78g'$$

Thus, if $g' \leq g(n) \leq n^{1/r(n)} < n^{1/k}$ (note $k < r(n)$), then the number n_0 of vertices in the graph G_0 must be bounded by $78n^{1/k}$. In this case, step 4.1 solves the problem in time $O(n_0^{k_0+1}) = O((n^{1/k})^k) = O(n)$. On the other hand, if G_0 has more than $78n^{1/k}$ vertices, then step 4.2 concludes correctly that the genus of the input graph G is larger than $g(n)$.

In conclusion, the algorithm **DS-FPT** solves the DOMINATING SET problem on graphs of genus bounded by $g(n)$ in time $O(2^{\overline{r}(k)} + 20^k + n)$, and the problem is fixed parameter tractable. □

We point out that the techniques used in Theorem 2.6 are simpler, more uniform, and derive much stronger results compared to the previous research, which was only valid for graphs of genus bounded by a constant [20]. Also, similarly to the algorithm **IS-FPT**, the algorithm **DS-FPT** does not have to know whether the input graph has minimum genus bounded by $g(n)$. For any graph of minimum genus bounded by $g(n)$, the algorithm will definitely derive a correct conclusion.

Theorem 2.7 *The* DOMINATING SET *problem on graphs of genus bounded by* $g(n)$ *is* $W[2]$*-complete if* $g(n) = n^{\Omega(1)}$.

PROOF. Since DOMINATING SET is $W[2]$-complete [19], it will suffice to show that DOMINATING SET on general graphs is fpt-reducible to the problem on graphs of genus bounded by $g(n)$. Since $g(n) = n^{\Omega(1)}$, we can assume that $g(n) \geq n^c$, where c is a fixed constant.

Let G_1 be an arbitrary graph of n_1 vertices and genus g_1. As we indicated in the proof of Theorem 2.2, $g_1 \leq n_1^2$. We construct a new graph G_2, which is the graph G_1, plus $n_1^{2/c} - n_1$ new vertices u, v, and v_i, $i = 1, 2, \ldots, n_1^{2/c} - n_1 - 2$, where u has degree 2 and is connected to the vertex v and to an arbitrary vertex in the graph G_1, and $[v, v_i]$, $i = 1, 2, \ldots, n_1^{2/c} - n_1 - 2$, make a star centered at v. It is fairly easy to verify that the graph G_2 has $n_2 = n_1^{2/c}$ vertices and genus $g_2 = g_1$, and that the graph G_1 has a dominating set of k_1 vertices if and only if the graph G_2 has a dominating set of $k_2 = k_1 + 1$ vertices. Since c is a constant, the reduction from (G_1, k_1) to (G_2, k_2) is an fpt-reduction. Moreover, since $g_2 = g_1 \leq n_1^2$, we have $g_2 \leq n_2^c \leq g(n_2)$. Therefore, (G_2, k_2) is an instance for DOMINATING SET on graphs of genus bounded by $g(n)$. This reduction proves that DOMINATING SET on graphs of genus bounded by $g(n)$ is $W[2]$-complete. □

Combining Theorem 2.6 and Theorem 2.7, we derive the following tight result.

Corollary 2.8 *Assuming $FPT \neq W[2]$, the* DOMINATING SET *problem on graphs of genus bounded by $g(n)$ is fixed parameter tractable if and only if $g(n) = n^{o(1)}$.*

3 Genus and subexponential time complexity

We say that a graph problem is solvable in *sublinear exponential time* (or shortly *subexponential time*) if it can be solved in time $2^{o(n)}$ on graphs of n vertices. Very few NP-hard graph problems are known to be solvable in subexponential time. Lipton and Tarjan used their planar graph separator theorem to show that a class of NP-hard planar graph problems, including VERTEX COVER, INDEPENDENT SET, and DOMINATING SET, are solvable in subexponential time [30]. They also described how their results can be extended to graphs of constant genus [30]. Recently, deriving lower bounds on the precise complexity of NP-hard problems has been attracting more and more attention [8, 27, 10, 11]. In particular, Impagliazzo, Paturi, and Zane [27] introduced the concept of SERF-reduction and showed that many well-known NP-hard problems are SERF-complete for the class SNP [27, 31]. This implies that if any of these problems is solvable in subexponential time, then so are all problems in the class SNP, a consequence that seems quite unlikely.

In this section, we demonstrate how graph genus affects the subexponential time computability for VERTEX COVER, INDEPENDENT SET, and DOMINATING SET. Our algorithmic results in this section extend Lipton and Tarjan's results on planar graphs and graphs of constant genus [30], and our lower bound results refine Impagliazzo, Paturi, and Zane's results on general graphs [27].

Proposition 3.1 ([18]) *Let G be a graph of n vertices and genus g. There is a linear time algorithm that partitions the vertices of G into three sets A, B, C, such that no edge joins a vertex in A with a vertex in B, $|A|, |B| \leq n/2$, and $|C| \leq c_0\sqrt{(g+1)n}$, where c_0 is a fixed constant.*

Theorem 3.2 *The problems* VERTEX COVER, INDEPENDENT SET, *and* DOMINATING SET *on graphs of genus bounded $g(n)$ are solvable in subexponential time if $g(n) = o(n)$.*

PROOF. We first give a detailed description of our proof for DOMINATING SET. The idea is quite simple: we use Proposition 3.1 to partition the vertices of a given graph G into the three sets A, B, and C, and enumerate all possible situations for the set C. Each fixed situation for the set C splits the graph G into two separated subgraphs, induced essentially by the vertex sets A and B, respectively. Thus, we can recursively work on the two subgraphs independently. However, this must be done with care. In particular, in a given situation for the set C, if a vertex u in C is assigned to be not in the dominating set and u is not adjacent to any vertex in C that is assigned to be in the dominating set, then the vertex u must remain in the graph and a vertex in A or B and adjacent to u must be included in the dominating set in a later stage.

Thus, assuming recursively that a partial dominating set D has been constructed, our recursive algorithm classifies the vertices in the remaining graph G into five groups:

(1) dominating vertices, which are already included in the current D;
(2) dominated vertices, which should not be in D and are adjacent to vertices in the current D;
(3) white vertices, which are adjacent to vertices in the current D but are not yet decided whether to be in D;
(4) black vertices, which are not adjacent to any vertices in the current D and are also not yet decided whether to be in D;
(5) red vertices, which should not be in D but are not yet adjacent to any vertices in the current D.

The dominating vertices and dominated vertices will be removed from the graph. Thus, the remaining graph G consists of only black, red, and white vertices (initially, $D = \emptyset$ and all vertices in G are black). Such a graph G will be called a *BWR-graph*. A *BW-dominating set* D' in the BWR-graph G is a set of black and white vertices in G such that every vertex in G is either in D' or adjacent to a vertex in D' (thus, a minimum BW-dominating set for the initial graph will be a regular minimum dominating set for the graph). To construct a minimum BW-dominating set for the BWR-graph G, we use Proposition 3.1 to partition the vertices of G into the three vertex subsets A, B, and C. Then we consider all possible assignments on the vertices in the set C. Each vertex u in C has the following possible assignments:

- u is a white vertex. Then either u is in D or u is not in D;
- u is a red vertex. Then u must be dominated by a vertex in either C, or A, or B;
- u is a black vertex. Then either u is in D, or u is not in D, and hence must be dominated by a vertex in either C, or A, or B.

An assignment to the vertices in C can be as follows: each white vertex is assigned either "in-D" or "not-in-D", each red vertex is assigned either "in-A" or "in-B", and each black vertex is assigned either "in-D", "in-A", or "in-B". After this assignment, a white vertex will become either a dominating vertex (if it is "in-D") or a dominated vertex (if it is "not-in-D"); a red vertex adjacent to an "in-D" vertex in C will become a dominated vertex (in this case, the assignment to the red vertex is ignored); a red vertex not adjacent to any "in-D" vertex in C will become a red vertex and will be added to the set A or B (depending on whether it is an "in-A" or "in-B" vertex); an "in-D" black vertex will become a dominating vertex; a black vertex whose status is either "in-A" or "in-B" and is adjacent to an "in-D" vertex in C will become a dominated vertex; finally, an "in-A" black vertex (resp. an "in-B" black vertex) not adjacent to any "in-D" vertex in C will become a red vertex and will be added to the set A (resp. B).

Let the subgraphs induced by the updated vertex sets A and B be G_A and G_B, respectively (note that now A and B may contain some vertices that were originally in C). We then recursively work on the subgraphs G_A and G_B. The algorithm is formally presented in Figure 3.

We analyze the algorithm. Suppose the original input graph G_0 has n_0 vertices. Set $b_0 = c_0\sqrt{g(n_0)+1}$, where c_0 is the constant given in Proposition 3.1 (the bound b_0 is fixed for all recursive calls to the algorithm **DS-Solver**). Suppose that the input to the algorithm **DS-Solver** is a BWR-graph G of n vertices. If $\sqrt{n} < 6b_0$, then $n < 36c_0^2(g(n_0)+1) = O(g(n_0))$, and a brute force method can construct a minimum BR-dominating set for G in time $O(3^n) = O(3^{O(g(n_0))})$. If $|C| > b_0\sqrt{n}$, then C would contain more than $c_0\sqrt{(g(n_0)+1)n}$ vertices. By Proposition 3.1, the graph G would have genus larger than $g(n_0)$, which implies that the original input graph G_0 has genus larger than $g(n_0)$ (since G is a subgraph of G_0). Thus, the algorithm stops correctly.

ALGORITHM. DS-solver
Input: a BWR-graph G of n vertices, and a bound b_0
Output: a minimum BR-dominating set D of G

1. **if** $\sqrt{n} < 6b_0$ **then** solve the problem by a brute force method; Stop;
2. partition the vertices of G into the subsets A, B, C, as described in Proposition 3.1;
3. **if** $|C| > b_0\sqrt{n}$ **then** Stop("the genus exceeds the bound");
4. **for** each assignment to the vertices in C **do**
   let D be the set of vertices in C that are assigned "in-D";
   update the graph G and the sets A and B;
   construct the subgraphs G_A and G_B;
   recursively construct the minimum BR-dominating sets D_A for G_A and D_B for G_B;
   $D = D \cup D_A \cup D_B$;
5. output the smallest BR-dominating set constructed in step 4.

Figure 3: An algorithm solving DOMINATING SET

Thus, we have $\sqrt{n} \geq 6b_0$ and $|C| \leq b_0\sqrt{n}$. Since each vertex in C can get at most 3 different assignments, the total number of different assignments to the set C is bounded by $3^{|C|} \leq 3^{b_0\sqrt{n}}$. Since originally, $|A|, |B| \leq n/2$, and the updated sets A and B are the original sets A and B plus some vertices in C, each of the subgraphs G_A and G_B contains at most $n/2 + b_0\sqrt{n} \leq 2n/3$ vertices (note that $b_0 \leq \sqrt{n}/6$). This gives the following recurrence relation for the time complexity $T(n)$ of the algorithm **DS-Solver**:

$$\begin{aligned} T(n) \leq 3^{b_0\sqrt{n}} \cdot 2T(2n/3) \leq 3^{b_0\sqrt{n}+1}T(2n/3) \qquad & \text{if } \sqrt{n} \geq 6b_0 \\ T(n) = O(3^{O(g(n_0))}) \qquad & \text{if } \sqrt{n} < 6b_0 \end{aligned}$$

Solving this recurrence relation, we get $T(n) = O(3^{O(b_0\sqrt{n}+g(n_0))})$. In particular, if we let $n = n_0$ and replace b_0 by $c_0\sqrt{g(n_0)+1}$, we get

$$T(n_0) = O(3^{O(c_0\sqrt{g(n_0)+1}\cdot\sqrt{n_0}+g(n_0))}) = 3^{O(\sqrt{n_0 g(n_0)}+g(n_0))} \tag{4}$$

Thus, if $g(n_0) = o(n_0)$, then $T(n_0) = 2^{o(n_0)}$, and the algorithm **DS-Solver** solves the DOMINATING SET problem in subexponential time.

The subexponential time algorithms for VERTEX COVER and INDEPENDENT SET are similar, and actually simpler. For example, for VERTEX COVER, once we partition the input graph into three parts A, B, and C, each vertex u in C has only two possibilities: either in or not in the minimum vertex cover W. In case u is in W, we simply remove u from the graph; while in case u is not in W, all neighbors of u are forced to be in W, thus all neighbors of u, as well as u itself, can be removed from the graph. Therefore, no vertices in C will be added to the sets A and B, and each of the induced subgraphs G_A and G_B will have at most $n/2$ vertices. This fact will simplify the analysis of the algorithm to derive the subexponential time bound. We leave the detailed verification to the interested readers. □

Again we point out that our subexponential time algorithms for DOMINATING SET, VERTEX COVER, and INDEPENDENT SET work correctly without needing to know the precise genus value of the input graph. The algorithms either report correctly that the genus of the input graph exceeds the designated bound $g(n)$, or construct an optimal solution to the input graph.

Remark. After the publication of a preliminary version [14] of the current paper in 2003, there has been some further progress in this direction. Demaine *et al.* [16] developed an algorithm of running time $2^{O(g\sqrt{k}+g^2)}n^{O(1)}$ for the parameterized DOMINATING SET problem on graphs of genus bounded by g, which was further improved by Fomin and Thilikos [23] who presented an algorithm of running time $2^{O(\sqrt{kg}+g)}+n^{O(1)}$. Compared to the algorithm in [16], our algorithm in Theorem 3.2 is faster when the graph genus g is $\Omega(\sqrt{n})$. Compared to the algorithm in [23], the running time of our algorithm (see Equality (4)) is of the same order as that of the algorithm in [23] for the general version (i.e., the non-parameterized version) of the DOMINATING SET problem (since the parameter k can be of order $\Theta(n)$). Moreover, our algorithm seems much simpler (the algorithm in [23] uses the techniques of graph representativity and graph branch decomposition).

Theorem 3.3 *For any function $g(n) = \Omega(n)$, if any of* VERTEX COVER, INDEPENDENT SET, *and* DOMINATING SET *on graphs of genus bounded by $g(n)$ can be solved in subexponential time, then all problems in the class SNP can be solved in subexponential time.*

PROOF. Since $g(n) = \Omega(n)$, we assume $g(n) \geq cn$, where c is a fixed constant. Johnson and Szegedy [28] have shown that if INDEPENDENT SET on graphs of degree bounded by 3 is solvable in subexponential time then so is INDEPENDENT SET on general graphs, which, according to Impagliazzo, Paturi, and Zane [27], would imply that all problems in the class SNP are solvable in subexponential time. Therefore, for INDEPENDENT SET, it suffices to show that the problem on graphs of degree bounded by 3 is reducible to the problem on graphs of genus bounded by $g(n)$ via a reduction that preserves the order of the number of vertices.

Let n_1, m_1, and g_1 be the number of vertices, the number of edges, and the genus of a graph G_1 of degree bounded by 3. Then $m_1 \leq 3n_1/2$, and by the Euler Polyhedral Equation [25], $g_1 \leq (m_1 - n_1 + 1)/2 \leq (n_1 + 2)/4 \leq n_1/3$ for $n_1 \geq 6$. If $c \geq 1/3$, then G_1 is already a graph of genus bounded by $cn_1 \leq g(n_1)$. Thus, we assume $c < 1/3$. We perform the following operation on the graph G_1. Pick any edge in G_1, and subdivide the edge by two degree-2 vertices. The resulting graph G' has $n_1 + 2$ vertices and the same genus g_1. Moreover, it can be proved [13, 15] that from any maximum independent set of G', a maximum independent set of G_1 can be constructed in linear time. Therefore, if we apply this edge subdivision operation $\lceil n_1/(6c) - n_1/2 \rceil$ times on the graph G_1, we get a graph G_2 of n_2 vertices and genus $g_2 = g_1$, where $n_1/(3c) \leq n_2 \leq (3c + 1)n_1/(3c)$. Now since $g_2 = g_1 \leq n_1/3 \leq cn_2$, the graph G_2 of n_2 vertices has genus bounded by $g(n_2)$. The reduction is completed by observing that $n_2 = O(n_1)$.

The theorem also holds for VERTEX COVER since INDEPENDENT SET can be reduced to VERTEX COVER using the same graph [24]. For DOMINATING SET, the theorem follows from the following facts: (1) VERTEX COVER on graphs of degree bounded by 3 can be reduced to DOMINATING SET on graphs of degree bounded by 6 [24]; and (2) subdividing an edge by three degree-2 vertices increases the minimum dominating set size by 1 [15] and does not change the graph genus. With these facts, the proof proceeds in a similar fashion to that for INDEPENDENT SET. We leave the details to interested readers. □

The class SNP [31] contains many well-known NP-hard problems, including k-SAT, k-COLORABILITY, k-SET COVER, VERTEX COVER, and INDEPENDENT SET [27]. It is commonly believed that it is unlikely that all problems in SNP are solvable in subexponential time. Based on this, and combining Theorem 3.2 and Theorem 3.3, we have the following tight results.

Corollary 3.4 *Assuming that not all the problems in SNP are solvable in subexponential time, the* VERTEX COVER, INDEPENDENT SET, *and* DOMINATING SET *problems on graphs of genus bounded by $g(n)$ are solvable in subexponential time if and only if $g(n) = o(n)$.*

4 Genus and approximability

We briefly review the related concepts and refer the readers to [6, 24] for more details. An *optimization problem* Q is either a *maximization* or a *minimization* problem. Each instance x of Q is associated with a set of *solutions* and each solution y for x is associated with a value $f(x, y)$. For a given instance x in Q, the objective is to find a solution with the maximum value $\max(x)$ (if Q is a maximization problem) or the minimum value $\min(x)$ (if Q is a minimization problem). An *approximation algorithm* A for Q is an algorithm that for each instance x of Q constructs a solution $A(x)$ for x. We say that the *approximation ratio* of the algorithm A is bounded by r if for all instances x of Q, we have $\max(x)/f(x, A(x)) \leq r$ (if Q is a maximization problem) or $f(x, A(x))/\min(x) \leq r$ (if Q is a minimization problem). We say that an optimization problem Q has a *polynomial time approximation scheme*, shortly *PTAS*, if for any constant $\epsilon > 0$, the problem Q has a polynomial time approximation algorithm whose approximation ratio is bounded by $1+\epsilon$. It is well-known that VERTEX COVER, INDEPENDENT SET, and DOMINATING SET on planar graphs have PTAS [7, 30].

Proposition 4.1 ([18]) *There is an $O(n \log g)$ time algorithm that for a given graph G of n vertices and genus g constructs a subset Z of at most $c\sqrt{gn \log g}$ vertices, where c is a fixed constant, such that removing the vertices in Z from G results in a planar graph.*

The algorithm in Proposition 4.1 does not need to know the genus of the input graph [18].

Theorem 4.2 *The* INDEPENDENT SET *problem on graphs of genus bounded by $g(n)$ has a PTAS if $g(n) = o(n/\log n)$.*

PROOF. Let $g(n) \leq n/(r(n) \log n)$, where $r(n)$ is a nondecreasing and unbounded function. Our PTAS for INDEPENDENT SET works as follows: for a given graph G of n vertices, we use the algorithm in Proposition 4.1 to construct the vertex subset Z (this can be done in time $O(n \log n)$ even when the genus of G is larger than $g(n)$). If the number z_0 of vertices in Z is larger than $c\sqrt{g(n)n \log g(n)})$, then we know that the input graph G has genus larger than $g(n)$ and we stop. Otherwise, the graph G_1 obtained by deleting the vertices in Z from the graph G is a planar graph. We apply any known PTAS algorithm (e.g., those given in [7, 30]) to construct an independent set I_1 for the graph G_1. We simply output I_1 as a solution to the original graph G.

It is obvious that this is a polynomial time approximation algorithm for INDEPENDENT SET on graphs of genus bounded by $g(n)$. What left is to analyze the approximation ratio of the algorithm. Because $g(n) \leq n/(r(n) \log n))$, the number of vertices z_0 in Z is such that $z_0 \leq c\sqrt{g(n)n \log g(n)}) \leq cn/\sqrt{r(n)}$. Let $n_1 = n - z_0$ be the number of vertices in the graph G_1. Let α and α_1 be the sizes of a maximum independent set in the graphs G and G_1, respectively. Then $\alpha_1 \leq \alpha \leq \alpha_1 + z_0$. Because G_1 is a planar graph, by the Four-Color theorem [25], $\alpha_1 \geq n_1/4$.

Let $\alpha_1' = |I_1|$. Since the independent set I_1 is constructed by a PTAS on the planar graph G_1, $\alpha_1/\alpha_1' \leq 1+\epsilon$, where ϵ is the given error bound. Since the function $r(n)$ is nondecreasing and unbounded, there is a constant N_0 such that when $n \geq N_0$, we have

$$\frac{c}{4\sqrt{r(n)}} \leq \frac{1}{8} \qquad \text{and} \qquad \frac{8c(1+\epsilon)}{\sqrt{r(n)}} \leq \epsilon \tag{5}$$

From the first inequality, we get

$$\alpha_1' \quad \geq \quad \frac{\alpha_1}{1+\epsilon} \geq \frac{n_1}{4(1+\epsilon)} = \frac{n - z_0}{4(1+\epsilon)} \geq \frac{n - cn/\sqrt{r(n)}}{4(1+\epsilon)}$$

$$= \quad n \cdot \left(\frac{1}{4(1+\epsilon)} - \frac{c}{4(1+\epsilon)\sqrt{r(n)}} \right) \geq \frac{n}{8(1+\epsilon)} \tag{6}$$

Since $\alpha \leq \alpha_1 + z_0 \leq (1+\epsilon)\alpha_1' + cn/\sqrt{r(n)}$, combining this with (5) and (6), we get

$$\frac{\alpha}{\alpha_1'} \leq 1 + \epsilon + \frac{cn}{\alpha_1' \sqrt{r(n)}} \leq 1 + \epsilon + \frac{8cn(1+\epsilon)}{n\sqrt{r(n)}} \leq 1 + 2\epsilon$$

Thus, the algorithm is a PTAS for INDEPENDENT SET on graphs of genus bounded by $g(n)$. □

Again our PTAS for INDEPENDENT SET does not need to know whether the input graph meets the given genus bound.

Theorem 4.3 *Assuming P $\neq$ NP, then* INDEPENDENT SET *on graphs of genus bounded by $g(n)$ has no PTAS if $g(n) = \Omega(n)$.*

PROOF. The proof uses techniques similar to those in Theorem 3.3, so we only give an outline of it. It is known that INDEPENDENT SET on graphs of bounded degree is APX-complete [6], which means that a PTAS for it would imply P = NP [5]. Now a graph G_1 of n_1 vertices and of bounded degree has its genus bounded by $O(n_1)$. We can increase the number of vertices in G_1 without changing the graph genus by subdividing the edges in G_1 by degree-2 vertices (see the proof of Theorem 3.3). This will give a graph G_2 of n_2 vertices whose genus is bounded by $g(n_2)$ (note that $g(n) \geq cn$ for some constant c), and a PTAS for the graph G_2 would imply a PTAS for the graph G_1. In consequence, a PTAS for INDEPENDENT SET on graphs of genus bounded by $g(n)$ would imply a PTAS for the same problem on graphs of bounded degree, which would imply that P = NP. □

Theorem 4.2 seems unlikely to hold for VERTEX COVER and DOMINATING SET. In fact, we can prove the following theorem.

Theorem 4.4 *Unless P = NP,* VERTEX COVER *and* DOMINATING SET *on graphs of genus bounded by $g(n)$ have no PTAS if $g(n) = n^{\Omega(1)}$.*

PROOF. It is known that VERTEX COVER and DOMINATING SET on general graphs have no PTAS unless P = NP [5, 31]. Thus, it suffices to show how these problems on general graphs can be reduced to the ones on graphs of genus bounded by $g(n) = n^{\Omega(1)}$. The proof is very similar to that for Theorem 2.7, thus we only give an outline of it. Consider the DOMINATING SET problem. For a given general graph G_1 of n_1 vertices, by attaching to G_1 a very large star, we can construct a new graph G_2 of n_2 vertices, without changing the graph genus, such that the genus of the graph G_2 is bounded by $g(n_2)$, and that the domination numbers of the graphs G_1 and G_2 differ by exactly 1. Now a PTAS for the graph G_2 would imply a PTAS for the graph G_1. The theorem for VERTEX COVER can be proved using a similar construction. □

On the other hand, we can derive results similar to Theorem 4.2 for VERTEX COVER and DOMINATING SET on "kernelized" graphs. Polynomial time kernelization algorithms have become an interesting topic in the recent research on NP-hard problems [2, 13, 23]. It has been demonstrated [26] that improvement on approximating VERTEX COVER and DOMINATING SET on kernelized graphs will directly imply the same improvement on approximating the problems on general graphs. In the following, we discuss the impact of graph genus on the approximability of VERTEX COVER and DOMINATING SET on kernelized graphs.

We say that a graph G of n vertices is *kernelized for the* VERTEX COVER *problem* if the number of vertices in a minimum vertex cover for G is at least $n/2$. Polynomial time kernelization algorithms have been developed [13, 26]. For an arbitrary graph G, the algorithms construct a kernelized graph G', where a vertex cover C' for the graph G' gives directly a vertex cover C for the graph G that preserves the approximation ratio (that is, the ratio of C to an optimal solution of G is not worse than the ratio of C' to an optimal solution of G').

Theorem 4.5 *The* VERTEX COVER *problem on kernelized graphs of genus bounded by $g_1(n)$ has a PTAS if $g_1(n) = o(n/\log n)$. On the other hand, unless P = NP, the* VERTEX COVER *problem on kernelized graphs of genus bounded by $g_2(n)$ has no PTAS if $g_2(n) = \Omega(n)$.*

PROOF. The development of a PTAS for VERTEX COVER on graphs of genus bounded by $g_1(n) = o(n/\log n)$ is very similar to that for the PTAS for INDEPENDENT SET given in Theorem 4.2, except that for INDEPENDENT SET in Theorem 4.2, we used Four-Color theorem to derive a linear lower bound on the size of maximum independent sets for planar graphs, while for VERTEX COVER on kernelized graphs, the linear lower bound on the size of minimum vertex covers comes directly from the fact that the input graph is kernelized. To prove that VERTEX COVER has no PTAS on kernelized graphs of genus bounded by $g_2(n) = \Omega(n)$, we use the techniques given in the proof of Theorem 4.3, by observing that a graph obtained by applying the operations given in Theorem 3.3 (i.e., subdividing an edge by two degree-2 vertices [15]) on a kernelized graph is also kernelized. We leave the detailed verification to the interested readers. □

Very recently, a kernelization algorithm for DOMINATING SET has been proposed. For a given graph G, let $\delta(G)$ be the size of a minimum dominating set in the graph G and recall that $\gamma_{\min}(G)$ denotes the minimum genus of the graph G. Formin and Thilikos [23] proposed a polynomial time algorithm that reduces a given graph G to a graph G' such that $\delta(G) = \delta(G')$, and such that the number of vertices of G' is bounded by $c_0(\delta(G') + \gamma_{\min}(G'))$, where $c_0 > 4$ is a constant. Based on this result, we can introduce the following definition: we say that a graph G is *kernelized for the* DOMINATING SET *problem* if the number of vertices in G is bounded by $c_0(\delta(G) + \gamma_{\min}(G))$, where c_0 is the constant given in [23].

Theorem 4.6 *The* DOMINATING SET *problem on kernelized graphs of genus bounded by $g_1(n)$ has a PTAS if $g_1(n) = o(n/\log n)$. On the other hand, unless P = NP, the* DOMINATING SET *problem on kernelized graphs of genus bounded by $g_2(n)$ has no PTAS if $g_2(n) = \Omega(n)$.*

PROOF. We only sketch the proof, which is similar to that for Theorem 4.5. We leave the detailed verification to the interested reader.

The PTAS for DOMINATING SET on kernelized graphs of genus bounded by $g_1(n) = o(n/\log n)$ is obtained in a similar way to the PTAS for VERTEX COVER given in Theorem 4.5, with the lower bound on the size of minimum dominating sets coming from the kernelization. To prove that DOMINATING SET has no PTAS on kernelized graphs of genus bounded by $g_2(n) = \Omega(n)$, we note that graphs of degree bounded by 3 are necessarily kernelized since the size of a minimum dominating set in such a graph is at least $n/4$ – each vertex can dominate at most 3 other vertices in the graph. Moreover, the genus of such a graph is bounded by $O(n)$ [25]. Therefore, the assumed PTAS for DOMINATING SET on graphs of genus bounded by $g_2(n)$ would imply a PTAS for DOMINATING SET on graphs of degree bounded by 3, which is APX-complete [6]. This, in consequence, would imply P = NP. □

5 Final remarks

We have demonstrated how graph genus affects the computational complexity of the well-known NP-hard problems VERTEX COVER, INDEPENDENT SET, and DOMINATING SET in terms of the following complexity measures: the fixed parameter tractability, the subexponential time computability, and the polynomial time approximability. In most cases, we were able to derive a precise genus threshold that uniquely determines the computational complexity of the problems in terms of the complexity measures. Our algorithmic results significantly extend the previous research on the problems on planar graphs and on graphs of constant genus, while our complexity results refine the previous results on the problems and identify the "hardest graph instances" for the problems. It should be easy to see that our techniques and results can be extended to other NP-hard graph problems.

It is NP-hard to determine the minimum genus of a given graph [32]. However, it is interesting to point out that all the algorithms developed in this paper work correctly without needing to know whether the input graph exceeds the designated genus bound. Our algorithms either report correctly that the input graph exceeds the designated genus bound, or solve the problems correctly for the given graph. Our techniques seem to be useful for the study of other computational problems related to graph genus.

Our results on the fixed parameter tractability and on the subexponential time computability (sections 2 and 3) are tight. Our results on the polynomial time approximation schemes (section 4), however, have a gap between $o(n/\log n)$ and $\Omega(n)$ on the genus bound. According to [18], when the graph genus is $o(n)$, there is a set of $o(n)$ vertices whose removal results in a planar graph. However, no algorithm is known that efficiently constructs such a set. It should be interesting and seems to be possible to close the genus gap in section 4.

Our results show that a class of NP-hard graph problems, including some very well-known ones, becomes more tractable on lower genus graphs while becomes more intractable on higher genus graphs. It is interesting to compare our results to the results in [4], which shows that certain other NP-hard problems become more tractable on dense graphs, for which the graph genus is necessarily high. We notice that the problems studied in [4] are most graph cutting problems, such as MAX-CUT, and GRAPH-BISECTION, while problems studied in the current paper are vertex subset problems. A systematical study of the difference between these two kinds of NP-hard problems looks rather appealing.

References

[1] J. ALBER, H. FAN, M. FELLOWS, H. FERNAU, R. NIEDERMEIER, F. ROSAMOND, AND U. STEGE, Refined search tree technique for dominating set on planar graphs, *Lecture Notes in Computer Science* **2136** (MFCS'01), (2001), pp. 111-122.

[2] J. ALBER, M. FELLOWS, AND R. NIEDERMEIER, Polynomial time data reduction for dominating set, *Journal of the ACM* **51**, (2004), pp. 363-384.

[3] J. ALBER, H. FERNAU, AND R. NIEDERMEIER, Parameterized complexity: exponential speedup for planar graph problems, *Journal of Algorithms* **52**, (2004), pp. 26-56.

[4] S. Arora, D. Karger, and M Karpinski, Polynomial time approximation schemes for dense instances of NP-hard problems, *Proc. 27th Annual ACM Symp. on Theory of Computing* (STOC'95), (1995), pp. 284-293.

[5] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, Proof verification and hardness of approximation problems, *Journal of the ACM* **45**, (1998), pp. 501-555.

[6] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*, Springer-Verlag, Berlin Heidelberg, 1999.

[7] B. Baker, Approximation algorithms for NP-complete problems on planar graphs, *Journal of the ACM* **41**, (1994), pp. 153-180.

[8] L. Cai and D. Juedes, On the existence of subexponential parameterized algorithms, *Journal of Computer and System Sciences* **67**, (2004), pp. 789-807.

[9] J. Chen, Algorithmic graph embeddings, *Theoret. Comput. Sci.* **181**, (1987), pp. 247-266.

[10] J. Chen, B. Chor, M. Fellows, X. Huang, D. Juedes, I. Kanj, and G. Xia, Tight lower bounds for certain parameterized NP-hard problems, *Proc. 19th Annual IEEE Conference on Computational Complexity* (CCC'04), (2004), pp. 150-160.

[11] J. Chen, X. Huang, I. Kanj, and G. Xia, Linear FPT reductions and computational lower bounds, *Proc. 34th ACM Symposium on theory of Computing* (STOC'04), (2004), pp. 212-221.

[12] J. Chen, S. Kanchi, and A. Kanevsky, A note on approximating graph genus, *Information Processing Letters* **61**, (1997), pp. 317-322.

[13] J. Chen, I. Kanj, and W. Jia, Vertex cover: further observations and further improvements, *Journal of Algorithms* **41**, (2001), pp. 280-301.

[14] J. Chen, I. Kanj, L. Perkovic, E. Sedgwick, and G. Xia, Genus characterizes the complexity of graph problems: some tight results *Proc. 30th International Colloquium on Automata, Languages, and Programming* (ICALP'03), *Lecture Notes in Computer Science 2719*, (2003), pp. 845-856.

[15] J. Chen, I. Kanj, and G. Xia, A note on subexponential parameterized complexity, *DePaul University Research Report No.* **03-002**, (2003). Available at `http://www.cs.depaul.edu/research/technical.asp`.

[16] D. Demaine, F. Fomin, M. Hajiaghayi, D. Thilikos, Subexponential parameterized algorithms on graphs of bounded-genus and H-minor-free graphs, *Proc. 15th ACM-SIAM Symp. Discrete Algorithms* (SODA'04), (2004), pp. 830-839.

[17] D. Demaine, M. Hajiaghayi, D. Thilikos, Exponential speedup of fixed-parameter algorithms on $K_{3,3}$-free or K_5-free graphs, *Lecture Notes in Computer Science* **2518** (ISAAC'02), (2002), pp. 262-273.

[18] H. Djidjev and S. Venkatesan, Planarization of graphs embedded on surfaces, *Lecture Notes in Computer Science* **1017** (WG'95), (1995), pp. 62-72.

[19] R. Downey and M. Fellows, *Parameterized Complexity*, Springer-Verlag, New York, 1999.

[20] J. Ellis, H. Fan, and M. Fellows, The dominating set problem is fixed parameter tractable for graphs of bounded genus, *Journal of Algorithms* **52**, (2004), pp. 152-168.

[21] H. Fleischner, S. Földes, and S. Szeider, Remarks on the Concept of Robust Algorithm, *RUTCOR Research Report No.* **26-2001**, (2001). Available at `http://rutcor.rutgers.edu/~rrr/2001.html`.

[22] F. Fomin and D. Thilikos, Dominating sets in planar graphs: branch-width and exponential speed-up, *Proc. 14th ACM-SIAM Symp. Discrete Algorithms* (SODA'03), (2003), pp. 168-177.

[23] F. Fomin and D. Thilikos, Fast parameterized algorithms for graphs on surfaces: linear kernel and exponential speed-up, *Proc. 31th International Colloquium on Automata, Languages, and Programming* (ICALP'04), *Lecture Notes in Computer Science 3142*, (2004), pp. 581-592.

[24] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, 1979.

[25] J. Gross and T. Tucker, *Topological Graph Theory*, Wiley-Interscience, New York, 1987.

[26] D. Hochbaum, Efficient bounds for the stable set, vertex cover and set packing problems, *Discrete Applied Mathematics* **6**, (1983), pp. 243-254.

[27] R. Impagliazzo, R. Paturi, and F. Zane, Which problems have strongly exponential complexity? *Journal of Computer and System Sciences* **63**, (2001), pp. 512-530.

[28] D. Johnson and M. Szegedy, What are the least tractable instances of max. independent set? *Proc. 10th Ann. ACM-SIAM Symp. on Discrete Algorithms* (SODA'99), (1999), pp. 927-928.

[29] I. Kanj and L. Perkovic, Improved parameterized algorithms for planar dominating set, *Lecture Notes in Computer Science* **2420** (MFCS'02), (2002), pp. 399-410.

[30] R. Lipton and R. Tarjan, Applications of a planar separator theorem, *SIAM Journal on Computing* **9**, (1980), pp. 615-627.

[31] Papadimitriou and M. Yannakakis, Optimization, approximation and complexity classes, *Journal of Computer and System Sciences* **43**, (1991), pp. 425-440.

[32] C. Thomassen, The graph genus problem is NP-complete, *Journal of Algorithms* **10**, (1989), pp. 568-576.