Combinatorics, Probability and Computing (2014) 23, 1028–1056. © Cambridge University Press 2014
doi:10.1017/S0963548314000339

Approximately Counting Embeddings into Random Graphs[†]

MARTIN FÜRER[1§] and SHIVA PRASAD KASIVISWANATHAN[2]

[1]Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, 16802, USA
(e-mail: furer@cse.psu.edu)

[2]General Electric Research, San Ramon, CA, 94568, USA
(e-mail: kasivisw@gmail.com)

Received 17 March 2013; revised 18 September 2013; first published online 9 July 2014

Let H be a graph, and let $C_H(G)$ be the number of (subgraph isomorphic) copies of H contained in a graph G. We investigate the fundamental problem of estimating $C_H(G)$. Previous results cover only a few specific instances of this general problem, for example the case when H has degree at most one (the monomer–dimer problem). In this paper we present the first general subcase of the subgraph isomorphism counting problem, which is almost always efficiently approximable. The results rely on a new graph decomposition technique. Informally, the decomposition is a labelling of the vertices such that every edge is between vertices with different labels, and for every vertex all neighbours with a higher label have identical labels. The labelling implicitly generates a sequence of bipartite graphs, which permits us to break the problem of counting embeddings of large subgraphs into that of counting embeddings of small subgraphs. Using this method, we present a simple randomized algorithm for the counting problem. For all decomposable graphs H and all graphs G, the algorithm is an unbiased estimator. Furthermore, for all graphs H having a decomposition where each of the bipartite graphs generated is small and almost all graphs G, the algorithm is a fully polynomial randomized approximation scheme.

We show that the graph classes of H for which we obtain a fully polynomial randomized approximation scheme for almost all G includes graphs of degree at most two, bounded-degree forests, bounded-width grid graphs, subdivision of bounded-degree graphs, and major subclasses of outerplanar graphs, series-parallel graphs and planar graphs of large girth, whereas unbounded-width grid graphs are excluded. Moreover, our general technique can easily be applied to proving many more similar results.

2010 *Mathematics subject classification*: Primary 68W20
Secondary 68Q87

[†] A preliminary version of this paper appeared in the *12th International Workshop on Randomization and Computation* (RANDOM 2008).

[§] Research supported in part by NSF grant CCF-0964655 and NSF grant CCF-1320814.

1. Introduction

Given a *template* graph H and a *base* graph G, we call an injection φ between vertices of H and vertices of G an *embedding* of H into G if φ maps every edge of H into an edge of G. In other words, φ is an isomorphism between H and a subgraph (not necessarily induced) of G. Deciding whether such an injection exists is known as the subgraph isomorphism problem. Subgraph isomorphism is an important and general form of pattern matching. It generalizes many interesting graph problems, including Clique, Hamiltonian Path, Maximum Matching, and Shortest Path. This problem arises in application areas ranging from text processing to physics and chemistry [8, 3, 35, 29]. The general subgraph isomorphism problem is NP-complete, but there are various special cases which are known to be fixed-parameter tractable in the size of H [2].

In this work, we consider the related fundamental problem of counting the number of copies of a template graph in another graph. By a *copy* of H in G we mean any, not necessarily induced subgraph of G, isomorphic to H. In general the problem is #P-complete (introduced by Valiant [37]). The class #P is defined as $\{f : \exists$ a non-deterministic polynomial-time Turing machine M such that on input x, the computation tree of M has exactly $f(x)$ accepting leaves$\}$. Problems complete for this class are presumably very difficult, especially since Toda's result [36] implies that a call to a #P-oracle suffices to solve any problem in the polynomial hierarchy in polynomial time.

Fixed-parameter tractability of this counting problem has been well studied with negative results for exact counting [11] and positive results for some special cases of approximate counting [4]. In this paper, we are interested in the more general problem of counting copies of large subgraphs. Exact counting is possible for very few classes of non-trivial large subgraphs. A key example is perfect matchings in a planar graph [27]. A slightly different problem that is also solvable in polynomial time is counting the number of spanning trees in a graph. A few more problems, such as counting perfect matchings in a bipartite graph (or (0–1) permanent) [25], counting all matchings in a graph [24], counting labelled subgraphs of a given degree sequence in a bipartite graph [5], counting combinatorial quantities encoded by the Tutte polynomial in a dense graph [1], and counting Hamilton cycles in dense graphs [9], can be done approximately. But problems such as counting perfect matchings in general graphs are still open.

Since most of the other interesting counting problems are hopelessly hard to solve (in many cases even approximately) [22], we investigate whether there exists a *fully polynomial randomized approximation scheme* (henceforth, abbreviated as FPRAS) that works well for *almost all graphs*. The statement can be made precise as follows. Let G_n be a graph chosen uniformly at random from the set of all n-vertex graphs. We say that a predicate $\mathcal{P}$ holds for almost all graphs if $\Pr[\mathcal{P}(G_n) = \mathit{true}] \to 1$ as $n \to \infty$ (probability over the choice of a random graph). By FPRAS we mean a randomized algorithm that produces a result that is correct to within a relative error of $1 \pm \epsilon$ with high probability (*i.e.*, probability tending to 1). The algorithm must run in time $\mathrm{poly}(n, \epsilon^{-1})$, where n is the input size. We call a problem *almost always efficiently approximable* if there is a randomized polynomial-time algorithm producing a result within a relative error of $1 \pm \epsilon$ with high probability for almost all instances.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

Previous attempts at solving these kinds of problems have not been very fruitful. For example, even seemingly simple problems such as counting cycles in a random graph have remained open for a long time (also stated as an open problem in the survey by Frieze and McDiarmid [14]). In this paper we present new techniques that can handle not only simple graphs such as cycles but also major subclasses of more complicated graph classes such as outerplanar, series-parallel, planar, and so on.

The theory of random graphs was initiated by Erdős and Rényi [10]. The most commonly used models of random graphs are $\mathcal{G}(n,p)$ and $\mathbb{G}(n,m)$. Both models specify a distribution on n-vertex graphs with a fixed set of vertices. In $\mathcal{G}(n,p)$ each of the $\binom{n}{2}$ edges is added to the graph independently with probability p and $\mathbb{G}(n,m)$ assigns equal probability to all graphs with exactly m edges. Unless explicitly stated otherwise, the default model addressed in this paper is $\mathcal{G}(n,p)$.

There has been a lot of interest in using random graph models for analysing typical cases (beating the pessimism of worst-case analysis). Here, we mention some of these results relevant to our counting problem (see the survey of Frieze and McDiarmid [14] for more). One of the most well-studied problems is that of counting perfect matchings in graphs. For this problem, Jerrum and Sinclair [23] presented a simulation of a Markov chain that is almost always an FPRAS (extended to all bipartite graphs in [25]). Similar results using other approaches were obtained later in [12, 31, 6, 17]. Another well-studied problem is that of counting Hamiltonian cycles in random digraphs. For this problem, Frieze and Suen [15] obtained an FPRAS, and later Rasmussen [31] presented a simpler FPRAS. Subsequently, Frieze, Jerrum, Molloy, Robinson and Wormald [13] obtained similar results in random regular graphs. Randomized approximation schemes are also available for counting the number of cliques in a random graph [32]. However, there are no general results for counting copies of an arbitrary given graph in a random graph.

1.1. Our results and techniques

In this paper, we remedy this situation by presenting the first general subcase of the subgraph isomorphism counting problem that is almost always efficiently approximable. For achieving this result we introduce a new graph decomposition that we call an *ordered bipartite decomposition*. Informally, an ordered bipartite decomposition is a labelling of vertices such that every edge is between vertices with different labels and for every vertex all neighbours with a higher label have identical labels. The labelling implicitly generates a sequence of bipartite graphs and the crucial part is to ensure that each of the bipartite graphs is of small size. The size of the largest bipartite graph defines the *width* of the decomposition. The decomposition allows us to obtain general results for the counting problem which could not be achieved using the previous methods. It also leads to a relatively simple and elegant analysis. We will show that many graph classes have such a decomposition, while at the same time many simple small graphs (such as triangles) may not possess a decomposition.

The actual algorithm itself is based on the following simple sampling idea (known as importance sampling in statistics): let $\mathcal{S} = \{x_1, \ldots, x_z\}$ be a large set whose cardinality we want to estimate. Assume that we have a randomized algorithm ($\mathcal{A}$) that picks each element x_i with non-zero known probability p_i. Then, Algorithm Count (Figure 1)

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

```
ALGORITHM COUNT($\mathcal{S} = \{x_1, \dots, x_z\}$,$\mathcal{A}$)
$t \leftarrow \mathcal{A}(\mathcal{S})$ (let Algorithm $\mathcal{A}$ return $x_i$ with probability $p_i > 0$ for all $i \in \{1, \dots, z\}$ with $\sum_{i=1}^{z} p_i \leqslant 1$)
If $t = x_i$ for $i \in \{1, \dots, z\}$, then $Z = 1/p_i$
Else $Z = 0$
Output $Z$
```

Figure 1. Estimator for the cardinality of $\mathcal{S}$.

produces an estimate for the cardinality of $\mathcal{S}$. The following proposition shows that the estimate is unbiased, *i.e.*, $\mathbb{E}[Z] = |\mathcal{S}|$.

Proposition 1.1. *Algorithm Count (Figure 1) is an unbiased estimator for the cardinality of $\mathcal{S}$.*

Proof. It suffices to show that each element x_i has an expected contribution of 1 towards $|\mathcal{S}|$. This holds because on picking x_i (an event that happens with probability p_i), we set Z to the inverse probability of this event happening. Therefore,

$$\mathbb{E}[Z] = \sum_i p_i \cdot \frac{1}{p_i} = |\mathcal{S}|. \qquad \square$$

Similar schemes of counting have previously been used by Hammersley [19] and Knuth [28] in other settings. This scheme was used by Rasmussen for approximating the permanent of a (0–1) matrix [31], and later for approximately counting cliques in a graph [32]. A variant of this scheme has also been used by the authors to provide a near linear-time algorithm for counting perfect matchings in random graphs [16, 17]. However, this is the first generalization of this simple idea to the general problem of counting graph embeddings. Another nice feature of such schemes is that they also seem to work well in practice [34].

Our randomized algorithm will try to embed H into G. If the algorithm succeeds in finding an embedding of H in G, it outputs the inverse probability of finding this embedding. The challenging task here is not only to ensure that each embedding of H in G has a positive probability of being found but also to pick each embedding with approximately equal probability to obtain a low variance. For this purpose, the algorithm considers an increasing sequence of subgraphs $\bar{H}_1 \subset \bar{H}_2 \subset \cdots \subset \bar{H}_\ell = H$ of H. The algorithm starts by randomly picking an embedding of $\bar{H}_1$ into G, then randomly an embedding of $\bar{H}_2$ into G containing the embedding of $\bar{H}_1$ and so on. It is for defining the increasing sequence of subgraphs that our decomposition is useful.

The algorithm is always an unbiased estimator for $C_H(G)$. The decomposition provides a natural sufficient condition for the class of algorithms based on the principle of Algorithm Count to be an unbiased estimator. Additionally, if the base graph is a random graph from $\mathcal{G}(n,p)$ with constant p and if the template graph has an ordered bipartite decomposition of bounded width, we show that the algorithm is an FPRAS. The interesting case of the result is when $p = 1/2$. Since the $\mathcal{G}(n,1/2)$ model assigns a uniform distribution over all graphs of n given vertices, an FPRAS (when the base graph is from $\mathcal{G}(n,1/2)$) can be

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

interpreted as an FPRAS for almost all base graphs. This result is quite powerful because now to prove that the number of copies of a template graph can be well approximated for most graphs G, one just needs to show that the template graph has an ordered bipartite decomposition of bounded width.

The latter half of the paper is devoted to showing that a lot of interesting graph classes naturally have an ordered bipartite decomposition of bounded width. Let $\mathcal{C}_k$ denote a cycle of length k. If a graph H does not have a subgraph isomorphic to $\mathcal{C}_k$, then we say H is $\mathcal{C}_k$-free.[1] In this paper, we show that graphs of degree at most two, bounded-degree forests, bounded-width grid (lattice) graphs,[2] subdivision of bounded-degree graphs,[3] bounded-degree outerplanar graphs which are $\mathcal{C}_3$-free, bounded-degree series-parallel graphs which are both $\mathcal{C}_3$- and $\mathcal{C}_5$-free,[4] and planar graphs of girth at least 16 have an ordered bipartite decomposition of bounded width. Using this we obtain the following result (proved in Theorems 3.7 and 4.1).

Theorem 1.2 (main result[5]). *Let H be a connected graph from one of the following graph classes: graphs of degree at most two, bounded-degree trees, bounded-width grid graphs, subdivision of bounded-degree graphs, bounded-degree $\mathcal{C}_3$-free outerplanar graphs, bounded-degree $[\mathcal{C}_3, \mathcal{C}_5]$-free series-parallel graphs, or bounded-degree planar graphs of girth at least 16. Then there exists an FPRAS for estimating the number of copies of H in $G \in \mathcal{G}(n, p)$ for constant p.*

Even when restricted to graphs of degree at most two, this theorem recovers most of the older results. It also provides simpler, unified proofs for (some of) the results in [12, 31, 6, 15]. For example, to count matchings of cardinality k, one could use a template consisting of k disjoint edges. Similarly, to count all cycles of length k the template is a cycle of that length. By varying k and boosting the success probability, the algorithm can easily be extended to count all matchings or all cycles. This provides the first FPRAS for counting all cycles in a random graph (solving an open problem of Frieze and McDiarmid [14]).

For template graphs coming from the other classes, our result supplies the first efficient randomized approximation scheme for counting copies of them in almost all base graphs. For example, it was not known earlier how to even obtain an FPRAS for counting the number of copies of a given bounded-degree tree in a random graph. For the simpler graph classes the decomposition follows quite straightforwardly, but for graph classes such as subdivision, outerplanar, series-parallel, and planar, constructing the decomposition requires several new combinatorial/algorithmic ideas. Even though our techniques can be

[1] This is a weaker definition than the notion of minor-free graphs used commonly in the graph theory literature [7].

[2] The *width* of an $n_1 \times n_2$ grid graph is $\min\{n_1, n_2\}$. It should not be confused with the width of a ordered bipartite decomposition.

[3] Given a graph, a subdivision graph is obtained by inserting at least one new vertex in each edge of the graph. See Section 4.2.

[4] Denoted henceforth as $[\mathcal{C}_3, \mathcal{C}_5]$-free.

[5] The proof of this theorem follows by combining Theorems 3.8 and 4.1.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

extended to other interesting graph classes, we conclude by showing that our techniques cannot be used to count the copies of an unbounded-width grid graph in a random graph.

Organization. In Section 2 we review some useful definitions. In Section 3 we define the ordered bipartite decomposition, and use that to obtain an FPRAS for counting copies of a graph in a random graph. Section 4 shows that many graph classes have an ordered bipartite decomposition of bounded width, whereas in Section 5 we show that an unbounded-width grid graph does not have this property. We conclude in Section 6.

2. Definitions and notation

Definition 2.1 (fully polynomial randomized approximation scheme (FPRAS)). Let Q be some function from the set of input strings Σ^* to natural numbers. A fully polynomial randomized approximation scheme for Q is a randomized algorithm that takes input $x \in \Sigma^*$ and an accuracy parameter $\epsilon \in (0,1)$ and outputs a number Z (a random variable depending on the coin tosses of the algorithm) such that,

$$\Pr[(1-\epsilon)Q(x) \leqslant Z \leqslant (1+\epsilon)Q(x)] \geqslant 3/4,$$

and runs in time polynomial in $|x|$, ϵ^{-1}. The success probability can be boosted to $1-\delta$ by running the algorithm $O(\log \delta^{-1})$ times and taking the median [26].

Graph notation. Throughout this paper, we use G to denote a base random graph on n vertices. The graph H is the template whose copies we want to count in G. We can assume without loss of generality that the graph H also contains n vertices, otherwise we just add isolated vertices to H. The number of isomorphic images remains unaffected. Let $\triangle = \triangle(H)$ denote the maximum degree of H.

For a graph F, we use V_F to denote its vertex set and E_F to denote its edge set. Furthermore, we use $v_F = |V_F|$ and $e_F = |E_F|$ for the number of vertices and edges. For a subset S of vertices of F, $N_F(S) = \{v \in V_F - S : \exists u \in S \text{ such that } (u,v) \in E_F\}$ denotes the neighbourhood of S in F. $F[S]$ denotes the subgraph of F induced by S.

Automorphisms are edge-respecting permutations on the set of vertices, and the set of automorphisms form a group under composition. For a graph H, we use $\mathrm{aut}(H)$ to denote the size of its automorphism group. For a bounded-degree graph H, $\mathrm{aut}(H)$ can be evaluated in polynomial time [30].

We use $C_H(G)$ to denote the number of copies of H in G. Let $L_H(G) = C_H(G) \cdot \mathrm{aut}(H)$ denote the number of embeddings (or labelled copies) of H in G. For a random graph G, we will be interested in quantities $\mathbb{E}[C_H(G)^2]$ and $\mathbb{E}[C_H(G)]^2$.

Most of the other graph-theoretic concepts that we use (such as planarity) are covered in standard text books (see, *e.g.*, [7]), and we describe them as needed.

Randomization. Our algorithm is randomized. The output of the algorithm is denoted by Z, which is an unbiased estimator of $C_H(G)$, *i.e.*, $C_H(G) = \mathbb{E}_{\mathcal{A}}[Z]$ (expectation over the coin tosses of the algorithm). As the output of our algorithm depends on both the input graph, and the coin tosses of the algorithm, we use expressions such as $\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z]]$. Here, the inner expectation is over the coin tosses of the algorithm, and the outer expectation

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

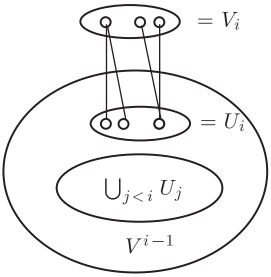

Figure 2.

is over the graphs of $\mathcal{G}(n,p)$. Note that $\mathbb{E}_{\mathcal{A}}[Z]$ is a random variable defined on the set of graphs.

3. Approximation scheme for counting copies

We define a new graph decomposition technique which is used for embedding the template graph into the base graph. As stated earlier, our algorithm for embedding works in stages and our notion of decomposition captures this idea.

Definition 3.1 (ordered bipartite decomposition). An ordered bipartite decomposition of a graph $H = (V_H, E_H)$ is a sequence $V_1, \ldots, V_\ell$ of subsets of V_H such that:

① $V_1, \ldots, V_\ell$ form a partition of V_H,
② each of the V_i (for $i \in [\ell] = \{1, \ldots, \ell\}$) is an independent set in H,
③ $\forall v\, \exists j$ such that $v \in V_i$ implies $N_H(v) \subseteq \left(\bigcup_{k<i} V_k\right) \cup V_j$.

Property ③ just states that if a neighbour of a vertex $v \in V_i$ is in some V_j $(j > i)$, then all other neighbours of v which are not in $V_1 \cup \cdots \cup V_{i-1}$, are in V_j. Property ③ will be used in the analysis for random graphs to guarantee that in every stage, the base graph used for embedding is still random with the original edge probability.

Let $V^i = \bigcup_{j \leqslant i} V_j$. Define

$$U_i = N_H(V_i) \cap V^{i-1}.$$

U_i is the set of neighbours of V_i in $V_1 \cup \cdots \cup V_{i-1}$. Define H_i to be the subgraph of H induced by $U_i \cup V_i$. Let E_{H_i} denote the edge set of graph H_i.

Definition 3.2 (width of ordered bipartite decomposition). Let $V_1, \ldots, V_\ell$ be the ordered bipartite decomposition of a graph $H = (V_H, E_H)$. Let U_i be the set of neighbours of V_i in $V_1 \cup \cdots \cup V_{i-1}$. Define H_i to be the subgraph of H induced by $U_i \cup V_i$. The *width* of an ordered bipartite decomposition of H is the number of edges (size) in the largest H_i. For an illustration, see Figure 2.

The U_i will play an important role in our analysis. Note that given a U_j, its corresponding V_j has the property that $V_j \supseteq N_H(U_j) - V^{j-1}$. Hereafter, when the context

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

is clear, we just use *decomposition* to denote an ordered bipartite decomposition. In general, the decomposition of a graph need not be unique. The following lemma describes some important consequences of the decomposition.

Lemma 3.3. *Let $V_1, \ldots, V_\ell$ be a decomposition of a graph $H = (V_H, E_H)$. Then the following assertions are true.*

(1) *Each of the U_i is an independent set in H (H_i is a bipartite graph).*
(2) *The edge set E_H is partitioned into $E_{H_1}, \ldots, E_{H_\ell}$.*

Proof. For part (1), assume otherwise. Let (u,v) be an edge in H with both $u, v \in U_i$. Let u appear in some V_j $(j < i)$ and v appear in some V_k $(k < i)$. Property ② implies that $j \neq k$. Assume without loss of generality that $j < k$. Property ③ implies there exists no vertex $w \in N_H(u)$ such that $w \in V_i$. Therefore $u \notin U_i$, a contradiction. In addition, since each of the U_i and V_i is an independent set, each of the graph H_i is bipartite.

For part (2), first note that due to properties ① and ③, the U_i are pairwise disjoint (but they do not necessarily form a partition). Therefore the E_{H_i} are also pairwise disjoint. Now since for every edge (u,v) there exist a j,k such that $u \in U_j$ and $v \in V_k$ and without loss of generality $j < k$, then $u \in U_k$ and $(u,v) \in E_{H_k}$. Thus, $E_{H_1}, \ldots, E_{H_\ell}$ form a partition of E_H. □

Every graph has a trivial decomposition satisfying properties ① and ②, but the situation changes if we add property ③ ($\mathcal{C}_3$ is the simplest graph which has no decomposition). Every bipartite graph, however, has a simple decomposition, but not necessarily of bounded width. Note that the bipartiteness of H is a sufficient condition for it to have an ordered bipartite decomposition, but not a necessary one.

We will primarily be interested in cases where the decomposition is of bounded width. This can only happen if $\triangle$ is a constant. In general, if $\triangle$ grows as a function of n, no decomposition could possibly have a bounded width ($\triangle/2$ is always a trivial lower bound for the width of a decomposition). The size of the parameter ℓ is not important in our analysis.

Algorithm for counting embeddings. The input to Algorithm Embeddings (Figure 3) is the template graph H together with its decomposition and the base graph G. The algorithm tries to construct an injection φ between the vertices of H (V_H) and G (V_G).

V_i represents the set of vertices of H which get embedded into G during the ith stage, and the already constructed mapping of U_i is used to achieve this. For a subset of vertices $S \subseteq V_H$, $\varphi(S)$ denotes the image of S under φ. If $X > 0$ (X is defined in Algorithm Embeddings), then the function φ represents an embedding of H in G (a consequence of properties ① and ②), and the output X represents the inverse probability of this event happening. Since every embedding has a positive probability of being found, X is an unbiased estimator for the number of embeddings of H in G (Proposition 1.1), and Z is an unbiased estimator for the number of copies of H in G.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

```
ALGORITHM EMBEDDINGS$(G, H)$
Initialize $X \leftarrow 1$; $\mathrm{Mark}(0) \leftarrow \emptyset$; $\forall v \in V_H, \varphi(v) \leftarrow \emptyset$; $U_1 \leftarrow \emptyset$
Let $V_1, \dots, V_\ell$ denote an ordered bipartite decomposition of $H$
For $i \leftarrow 1$ to $\ell$ do
    Let $G_f \leftarrow G[V_G - \mathrm{Mark}(i-1)]$ ($G_f$ is the subgraph of $G$ used for embedding $H_i$)
    Compute $X_i$, the number of embeddings of $H_i$ in $G_f$, with the mapping of vertices in $U_i$ to the
            vertices in $V_G$ fixed as defined by $\varphi$
    If $X_i > 0$
        Pick an embedding $\psi$ uniformly at random from the above set of embeddings ($\psi$ is an
injection            from $V_i$ to $V_G$)
        Update $\varphi$ using $\psi$ as follows: $\forall v \in V_i, \varphi(v) \leftarrow \psi(v)$
    Else
        Set $Z$ to $0$ and terminate
    $X \leftarrow X \cdot X_i$
    $\mathrm{Mark}(i) \leftarrow \mathrm{Mark}(i-1) \cup \left(\bigcup_{v \in V_i} \varphi(v)\right)$
$Z \leftarrow X / \mathrm{aut}(H)$
Output $Z$
```

Figure 3. Algorithms for counting copies of graph H in G. In iteration i, φ is an injection from $\bigcup_{1 \leqslant j \leqslant i} V_j \subseteq V_H$ into V_G defining a partial embedding of H into G.

The actual procedure for computing the X_i is not very relevant to our results, but note that the X_i can be computed in polynomial time if H has a decomposition of bounded width. In this case Algorithm Embeddings runs in polynomial time.

Since Algorithm Embeddings is an unbiased estimator, use of Chebyshev's inequality implies that repeating the algorithm $O(\epsilon^{-2}\mathbb{E}_{\mathcal{A}}[Z^2]/\mathbb{E}_{\mathcal{A}}[Z]^2)$ times and taking the mean of the outputs results in a randomized approximation scheme for estimating $C_H(G)$. The ratio $\mathbb{E}_{\mathcal{A}}[Z^2]/\mathbb{E}_{\mathcal{A}}[Z]^2$ is commonly referred to as the *critical ratio*.

3.1. FPRAS for counting in random graphs

We now concentrate on showing that for random graphs the algorithm is an FPRAS. From here on, we abbreviate $C_H(G)$ as C. A few of the technical details in our proof are somewhat similar to previous applications of this sampling idea, such as that for counting perfect matchings [31, 17]. The simpler techniques in these previous results, however, are limited to handling one edge per stage (therefore, they work only when H is a matching). Algorithm Embeddings embeds a small-sized subgraph at every stage. The key to obtaining an FPRAS is to guarantee that the factor contributed to the critical ratio at every stage is very small (which is now involved because it is no longer a simple ratio of binomial moments as in [31, 17]). We then do a stage-by-stage analysis of the critical ratio to show that Algorithm Embeddings is an FPRAS.

The analysis will be done for a worst-case graph H under the assumption that the width of the decomposition of H is bounded by a universal constant w. Here, instead of investigating the critical ratio, we investigate the much simpler ratio $\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z^2]]/\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z]]^2$, which we call the *critical ratio of averages*. We use the second moment method to show that these two ratios (critical ratio and critical ratio of averages) are closely related. To establish this fact, we take a detour through the $\mathbb{G}(n, m)$ model. The ratio $\mathbb{E}[C^2]/\mathbb{E}[C]^2$ plays an important role here, and to bound it we use a recent result of Riordan [33].

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

The result (stated below) studies the related question of when a random graph G is likely to have a spanning subgraph isomorphic to H. Let $\triangle = \triangle(H)$ denote the maximum degree of H. The idea behind the following theorem is to use Markov's inequality to bound $\Pr[C = 0]$ in terms of $\mathbb{E}[C]$ and $\mathrm{Var}[C]$. The main thrust lies in proving that $\mathbb{E}[C^2]/\mathbb{E}[C]^2 = 1 + o(1)$.[6]

In the following, N is used to denote $\binom{n}{2}$. Let $\{H^{(n)}\}$ be a sequence of graphs, where $H^{(n)}$ has n vertices. For notational convenience, we will frequently write H for $H^{(n)}$. We say that H has a property $P^{(n)}$ when we implicitly assume that there is a sequence of graphs $\{H^{(n)}\}$, and for every n, the graph $H^{(n)}$ has property $P^{(n)}$. We do it similarly for G. An event holds with high probability (w.h.p.) if it holds with probability tending to 1 as $n \to \infty$.

Theorem 3.4 (Riordan [33], restated). *Let H be a graph on n vertices. Let $e_H = \alpha N = \alpha(n)N$, and let $p = p(n) \in (0, 1)$ with pN an integer. Suppose that the following conditions hold: $\alpha N \geqslant n - 1$, and $pN, (1 - p)\sqrt{n}, np^{\gamma}/\triangle^4 \to \infty$, where*

$$\gamma = \gamma(H) = \max_{3 \leqslant s \leqslant n} \{\max\{e_F : F \subseteq H, v_F = s\}/(s - 2)\}.$$

Then, w.h.p. a random graph $G \in \mathbb{G}(n, pN)$ has a spanning subgraph isomorphic to H. In general, $C = C_H(G)$ satisfies

$$\frac{\mathbb{E}[C^2]}{\mathbb{E}[C]^2} = 1 + o(1).$$

The quantity γ is closely related to twice the maximum average degree of a subgraph of H.

Remark. Riordan [33] establishes Theorem 3.4 using the second moment method, a technique that relies on a bound on the ratio between second moment and first moment square (which in this case is $\mathbb{E}[C^2]/\mathbb{E}[C]^2$). In [33], a bound on this ratio is obtained in the proof of Theorem 2.1 (this ratio is referred to as f in [33]).

The templates graph that we will be interested in are bounded-degree connected graphs. For a bounded-degree graph H, both $\triangle$ and γ are constants. Also, since the graph is connected, $\alpha N \geqslant n - 1$. Further, for us p is a constant (as we work with dense random graphs G). Therefore, the conditions of Theorem 3.4 are all satisfied.

Corollary 3.5. *Let H be a bounded-degree connected graph on n vertices. Then, w.h.p. a random graph $G \in \mathbb{G}(n, \Omega(n^2))$ satisfies $\mathbb{E}[C^2]/\mathbb{E}[C]^2 = 1 + o(1)$.*

[6] Since C is fairly tightly concentrated around its mean, a rudimentary approximation for C is just $\mathbb{E}[C] = n!p^{e_H}/\mathrm{aut}(H)$ (as $v_H = n$). However, this naive approach does not produce, for *any* $\epsilon > 0$, an $(1 \pm \epsilon)$-approximation for C (see, *e.g.*, [12, 15, 31, 32, 6]).

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

Corollary 3.5 with Chebyshev's inequality gives $\Pr[C \leqslant \beta\mathbb{E}(C)]$, for any constant $\beta < 1$, and tends to 0 as n tends to ∞. Using this and standard results on asymptotic equivalence between $\mathbb{G}(n,m)$ and $\mathcal{G}(n,p)$ models of random graphs (*e.g.*, see Proposition 1.12 of [21]) yields the following corollary. A similar analysis has been used in previous work [12, 15, 31, 32, 6].

Corollary 3.6. *Let H be a bounded-degree connected graph on n vertices. Let $\omega = \omega(n)$ be any function tending to ∞ as $n \to \infty$, and let p be a constant. Then, w.h.p. a random graph $G \in \mathcal{G}(n,p)$ satisfies $C \geqslant \mathbb{E}[C]/\omega$.*

Using the above result we investigate the performance of Algorithm Embeddings when G is a random graph. The proof idea is to break the critical ratio analysis of the large subgraph into a more manageable critical ratio analysis of small subgraphs.

Proposition 3.7. *Let H be an n-vertex connected graph with a decomposition of width w (a constant). Let Z be the output of Algorithm Embeddings, and let p be a constant. Then, w.h.p. for a random graph $G \in \mathcal{G}(n,p)$, the critical ratio $\mathbb{E}_{\mathcal{A}}[Z^2]/\mathbb{E}_{\mathcal{A}}[Z]^2$ is polynomially bounded in n.*

Proof. We first relate the critical ratio to the critical ratio of averages. As the estimator is unbiased, $\mathbb{E}_{\mathcal{A}}[Z] = C$. Therefore, from Corollary 3.6, w.h.p.

$$C = \mathbb{E}_{\mathcal{A}}[Z] = \frac{\mathbb{E}_{\mathcal{A}}[X]}{\operatorname{aut}(H)} \geqslant \frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]}{\omega \cdot \operatorname{aut}(H)}.$$

Squaring both sides, we have that w.h.p.

$$\mathbb{E}_{\mathcal{A}}[Z]^2 = \frac{\mathbb{E}_{\mathcal{A}}[X]^2}{\operatorname{aut}(H)^2} \geqslant \frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]^2}{\omega^2 \operatorname{aut}(H)^2}.$$

Note that $\mathbb{E}_{\mathcal{A}}[X]/\operatorname{aut}(H)$ refers to the expected output for fixed graph G, and the inequalities hold for almost all such graphs G, while $\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]/\operatorname{aut}(H)$ is the expected output for a random graph $G \in \mathcal{G}(n,p)$.

The numerator of the critical ratio of averages satisfies

$$\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z^2]] = \frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X^2]]}{\operatorname{aut}(H)^2}.$$

Using Markov's inequality,

$$\Pr\big[\mathbb{E}_{\mathcal{A}}[Z^2] \geqslant \omega\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z^2]]\big] \leqslant \frac{1}{\omega} \xrightarrow{n\to\infty} 0.$$

Using the above inequalities yields that w.h.p.

$$\frac{\mathbb{E}_{\mathcal{A}}[Z^2]}{\mathbb{E}_{\mathcal{A}}[Z]^2} = \frac{\mathbb{E}_{\mathcal{A}}[X^2]}{\mathbb{E}_{\mathcal{A}}[X]^2} \leqslant \omega^3 \frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z^2]]}{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[Z]]^2} = \omega^3 \frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X^2]]}{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]^2}.$$

Now, we just concentrate on bounding the critical ratio of averages, that is,

$$\frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X^2]]}{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]^2}.$$

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

Let $V_1, \ldots, V_\ell$ denote a decomposition of H of width w. In the bipartite graph H_i between the vertices of U_i and V_i with edge set E_{H_i}, let $e_i = |E_{H_i}|$, $v_i = |V_i|$, and $u_i = |U_i|$. Let $n_i = n - \sum_{j<i} v_j$. We will rely on the fact that all the H_i are of bounded size (their maximum defining the width of the decomposition).

Let $n_i' = n_i + u_i$. Let G_i be a random graph from $\mathcal{G}(n_i', p)$ with u_i distinguished vertices. Let $L_{H_i|U_i}(G_i)$ denote the number of embeddings of H_i in G_i where the mapping of the vertices in U_i to the distinguished vertices in G_i is fixed (given). The results do not depend on the mapping used for U_i. We abbreviate $L_{H_i|U_i}(G_i)$ by L_i.

First we investigate the numerator of the critical ratio of averages. Here we use the fact that

$$\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X^2]] = \mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_1^2]] \cdot \cdots \cdot \mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_\ell^2]].$$

The previous equality arises because at the ith stage the graph used for embedding H_i is from $\mathcal{G}(n_i', p)$, irrespective of the choices made over the first $(i-1)$ stages. This is guaranteed by property ③ of the decomposition, and in turn it allows us to perform a stage-by-stage analysis of the critical ratio.

Furthermore, $\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_i^2]] = \mathbb{E}[L_i^2]$ (as the graph is random, it does not matter which vertices U_i gets mapped to). Next we investigate the denominator of the critical ratio of averages. Here we use the fact that

$$\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]^2 = (n!p^{e_H})^2 = \mathbb{E}[L_1]^2 \cdot \cdots \cdot \mathbb{E}[L_\ell]^2.$$

Therefore, the ratio

$$\begin{aligned}\frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X^2]]}{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]^2} &= \frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_1^2]] \cdot \cdots \cdot \mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_\ell^2]]}{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_1]]^2 \cdot \cdots \cdot \mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_\ell]]^2}\\ &= \frac{\mathbb{E}[L_1^2] \cdot \cdots \cdot \mathbb{E}[L_\ell^2]}{\mathbb{E}[L_1]^2 \cdot \cdots \cdot \mathbb{E}[L_\ell]^2} = \prod_{i=1}^{\ell} \frac{\mathbb{E}[L_i^2]}{\mathbb{E}[L_i]^2}.\end{aligned}$$

To bound this expression we investigate the parameter $\mathrm{Var}[L_i]$.

Now consider a complete bipartite graph K_{u_i,n_i} with one side being the u_i distinguished vertices of G_i and the other side being the remaining (non-distinguished) vertices of G_i. Let $\mathcal{F}_{H_i|U_i}(K_{u_i,n_i})$ be the set of embeddings of H_i in K_{u_i,n_i}, where the mapping of the vertices in U_i to the distinguished vertices in K_{u_i,n_i} is fixed as in G_i (note that one side of K_{u_i,n_i} contains these distinguished vertices). For each embedding f from $\mathcal{F}_{H_i|U_i}(K_{u_i,n_i})$ define the indicator random variable $I_{f(H_i)} = \mathbf{1}[f(H_i) \subseteq G_i]$. For each $F \subseteq H_i$, let e_F be the number of edges in F, and let r_F be the number of vertices in F which belong to V_i. Now there are $\Theta(n_i^{2v_i - r_F})$ pairs (f, g) of embeddings of H_i in $\mathcal{F}_{H_i|U_i}(K_{u_i,n_i})$ with $f(H_i) \cap g(H_i)$ isomorphic $(\simeq)$ to F. In the following, we use $A \asymp B$ for $A = \Theta(B)$:

$$\begin{aligned}\mathrm{Var}[L_i] &= \sum_{f,g} \mathrm{Cov}[I_{f(H_i)}, I_{g(H_i)}]\\ &= \sum_{\substack{f,g\\ E_{f(H_i)} \cap E_{g(H_i)} \neq \emptyset}} \mathbb{E}[I_{f(H_i)} I_{g(H_i)}] - \mathbb{E}[I_{f(H_i)}]\mathbb{E}[I_{g(H_i)}]\end{aligned}$$

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

$$\begin{aligned}
&= \sum_{F\subseteq H_i, e_F>0} \sum_{\substack{f,g\\ f(H_i)\cap g(H_i)\simeq F}} \mathbb{E}[I_{f(H_i)}I_{g(H_i)}] - \mathbb{E}[I_{f(H_i)}]\mathbb{E}[I_{g(H_i)}]\\
&= \sum_{F\subseteq H_i, e_F>0} \sum_{\substack{f,g\\ f(H_i)\cap g(H_i)\simeq F}} p^{2e_i-e_F} - p^{2e_i}\\
&\asymp \sum_{F\subseteq H_i, e_F>0} n_i^{2v_i-r_F}(p^{2e_i-e_F} - p^{2e_i})\\
&= \sum_{F\subseteq H_i, e_F>0} \frac{n_i^{2v_i}p^{2e_i}}{n_i^{r_F}p^{e_F}}(1-p^{e_F})\\
&\asymp \sum_{F\subseteq H_i, e_F>0} \frac{\mathbb{E}[L_i]^2}{n_i^{r_F}p^{e_F}}(1-p^{e_F})\\
&\asymp \max_{F\subseteq H_i, e_F>0} \frac{\mathbb{E}[L_i]^2}{n_i^{r_F}p^{e_F}}(1-p^{e_F}).
\end{aligned}$$

The second equality (above) used the fact that random variables $I_{f(H_i)}$ and $I_{g(H_i)}$ are independent if $E_{f(H_i)} \cap E_{g(H_i)} = \emptyset$. The implicit constants in the above equivalences depend on the size of H_i (a constant), but are independent of n_i. The quantity

$$\max_{F\subseteq H_i, e_F>0} \frac{(1-p^{e_F})}{n_i^{r_F}p^{e_F}} = O(1/n_i) \ (r_F = 1, \text{ provides the maximum}).$$

Therefore, $\mathrm{Var}[L_i]/\mathbb{E}[L_i]^2 = O(1/n_i)$, implying $\mathbb{E}[L_i^2]/\mathbb{E}[L_i]^2 = 1 + O(1/n_i)$. If $e_i = 0$, then $\mathrm{Var}[L_i] = 0$, and $\mathbb{E}[L_i^2] = \mathbb{E}[L_i]^2$. Putting everything together, we obtain

$$\frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X^2]]}{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]^2} = \prod_{i=1}^{\ell} \frac{\mathbb{E}[L_i^2]}{\mathbb{E}[L_i]^2} \leqslant \prod_{i=1}^{\ell}\left(1 + \frac{c}{n_i}\right) = \prod_{i=1}^{\ell} \frac{n_i + c}{n_i},$$

for a constant c depending only on w and p. Since $n = n_1 > n_2 > \cdots > n_\ell$, $\prod_{i=1}^{\ell}(n_i + c)/n_i$ can be polynomially bounded (to $O(n^c)$) by a telescoping argument. Putting everything together, we get that w.h.p. for a random graph $G \in \mathcal{G}(n,p)$ the critical ratio $\mathbb{E}_{\mathcal{A}}[Z^2]/\mathbb{E}_{\mathcal{A}}[Z]^2$ is polynomially bounded in n. This completes the proof. □

Summarizing, we have the following result. If H has a decomposition of bounded width w, then for almost all graphs G, running Algorithm Embeddings $\mathrm{poly}(n)\epsilon^{-2}$ times and taking the mean of the outputs it generates results in an $(1 \pm \epsilon)$-approximation for C. Here, $\mathrm{poly}(n)$ is a polynomial in n depending on w and p. Since each run of Algorithm Embeddings also takes polynomial time (as H has a bounded-width decomposition), this is an FPRAS.

Theorem 3.8. *Let H be an n-vertex connected graph with a decomposition of width w (a constant). Then, there exists an FPRAS for estimating the number of copies of H in $G \in \mathcal{G}(n,p)$ for constant p.*

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

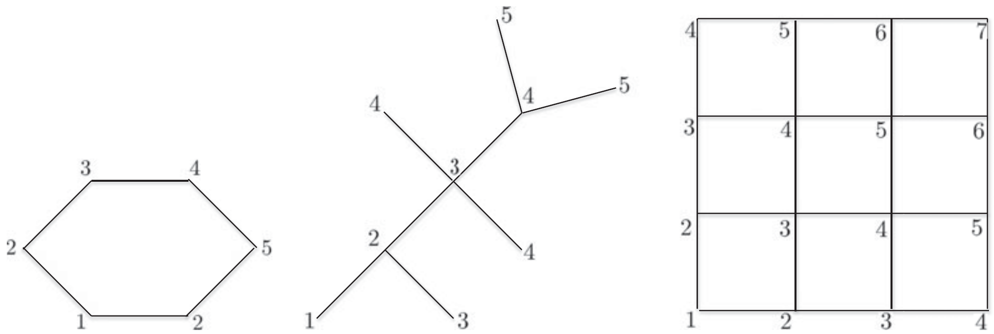

Figure 4. Decomposition of a cycle, tree, and grid. Vertices with label i constitute V_i. Neighbours of V_i with lower labels constitute U_i.

4. Graphs with ordered bipartite decomposition

We divide this section into subsections based on the increasing complexity of the graph classes. We will prove the following result in the remainder of this section.

Theorem 4.1.[7] *Let H be a graph from one of the following graph classes: graphs of degree at most two, forests, bounded-width grid graphs, subdivision graphs, $\mathcal{C}_3$-free outerplanar graphs, $[\mathcal{C}_3,\mathcal{C}_5]$-free series-parallel graphs, or planar graphs of girth at least* 16*. Then, there exists an ordered bipartite decomposition of H. Furthermore, if H has bounded degree, then the decomposition has bounded width.*

We concentrate on connected graphs H.[8] Let $\triangle$ be the maximum degree of any vertex in H. To construct the decomposition, the following definitions are useful:

$$U^i = \bigcup_{j\leqslant i} U_j, \quad V^i = \bigcup_{j\leqslant i} V_j \quad \text{and} \quad D^i = V^i - U^i.$$

All our decomposition algorithms proceed in steps, with step i creating the (U_i, V_i) pair.

4.1. Some easy graph classes

We start off by considering easy graph classes such as graphs of degree at most two (paths and cycles), trees, and grid graphs. Figure 4 illustrates some examples.

- *Paths.* Let H represent a path $(s_1,\ldots,s_{k+1})$ of length $k = k(n)$. Then the decomposition is $V_i = \{s_i\}$ for $1 \leqslant i \leqslant k+1$.
- *Cycles.* First consider the cycles of length four or greater. Let $s_1,\ldots,s_k$ be the vertices of a cycle H of length $k = k(n)$ enumerated in cyclic order. In the decomposition, $V_1 = \{s_1\}$, $V_2 = \{s_2, s_k\}$, and $V_i = \{s_i\}$ for $3 \leqslant i \leqslant k-1$. Cycles of length three (triangles) do not have a decomposition, but counting copies of triangles is easy (we describe

[7] The proof of this theorem follows by combining Propositions 4.2, 4.4, 4.7, 4.9, and 4.12.

[8] If H is disconnected, then a decomposition is obtained by combining the decomposition of all the connected components (in any order).

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

an algorithm to do so in Appendix A). This also completes the claim for graphs of degree at most two in Theorem 4.1.

- *Trees.* For a tree H, we have $V_1 = \{s_1\}$, where s_1 is any vertex in H. For $i \geqslant 2$, let U_i be any vertex from D^{i-1}, and then V_i is the set of neighbours of this vertex which are not in V^{i-1}. Intuitively, V_i is the set of children of the vertex in U_i, if one thinks of H as a tree rooted at s_1. The width of this decomposition is at most $\triangle$.
- *Grid graphs.* Let w_0 be the width of the grid graph H (for an $n_1 \times n_2$ grid graph the width is $\min\{n_1, n_2\}$). Set $V_1 = \{s_1\}$, where s_1 is any corner vertex in H. Later on, V_i is the set of all vertices which are at a lattice (Manhattan) distance i from s_1. Since for each i there are at most w_0 vertices at distance i from s_1, the sizes of the V_i are bounded if w_0 is bounded. Consequently, the width of this decomposition is bounded if w_0 is bounded. This construction also extends to higher-dimensional grid graphs.

Proposition 4.2. *Let H be a graph from one of the following graph classes: graphs of degree at most two, bounded-degree forests, or bounded-width grid graphs. Then, there exists an ordered bipartite decomposition of H with bounded width.*

4.2. Decomposition of subdivision graphs

A k-subdivision graph of a graph is obtained by inserting $k = k(n)$ new vertices in every edge, that is, by replacing each original edge by a path of length $k+1$. We relax this definition and say that a k-subdivision graph is the graph obtained by inserting at least one and at most k vertices in every edge. Let H be a k-subdivision graph of a graph F. We now show that H has a decomposition of width at most $\triangle$.

The main idea behind the decomposition is that as soon as a vertex v of F appears in some V_j, all vertices in $N_H(v)$ not in V^j are selected in V_{j+1}, *i.e.*, $v \in U_{j+1}$. The decomposition of H can be formally defined as

$$V_i = \begin{cases} \{s_1\}, \text{ where } s_1 \text{ is any vertex in } V_F & \text{if } i = 1, \\ N_H(a_i) - V^{i-1} & \text{if } i \geqslant 2 \text{ and } \{a_i\} = V_F \cap D^{i-1} \neq \emptyset, \\ N_H(b_i) - V^{i-1}, \text{ where } b_i \text{ is any vertex in } D^{i-1} & \text{otherwise.} \end{cases}$$

We now argue the correctness of the decomposition, for which the following lemma is useful.

Lemma 4.3. *There exists at most one vertex in $V_F \cap D^i$ for all i in the decomposition.*

Proof. Proof by induction on i. It is true by construction for $i = 1$. Assume by the inductive hypothesis that $V_F \cap D^{i-1}$ has at most one vertex. If there exists a vertex in $V_F \cap D^{i-1}$, then let a_i be this vertex. In this case, $N_H(a_i)$ does not contain any vertex from V_F (this follows as subdivision of F creates H). Otherwise, $b_i \notin V_F$, therefore, there is at most one vertex of V_F in V_i (again this follows because subdivision of F creates H). Therefore, in both cases, $|V_F \cap D^i| \leqslant 1$. □

Notice that the decomposition described above selects all vertices in H, and the vertices selected in any V_i are not selected in V^{i-1}, therefore, the V_i form a partition of V_H

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

(property ①). For property ② notice that if V_i is constructed using a_i (or s_1), then (by the subdivision graph construction) it is always the case that $N_H(a_i)$ is an independent set, and if it is constructed using b_i, then it has at most two neighbours that do not have an edge between them (again due to the subdivision graph construction). Property ③ is satisfied, as for the vertex a_i or b_i (or s_1) we select all its neighbours that are not in V^{i-1} together.

The width of this decomposition is at most $\triangle$ as the maximum degree of H is $\triangle$.

Proposition 4.4. *Let H be a subdivision of a bounded-degree graph. Then, there exists an ordered bipartite decomposition of H with bounded width.*

4.3. Decomposition of outerplanar graphs

In this section, we prove the decomposition property on outer planar graphs. A graph is outerplanar if it has a planar embedding such that all vertices are on the same face. Let H be a C_3-free outerplanar graph. The idea behind the decomposition is that vertices in U_i partition the outer face into smaller intervals, each of which can then be handled separately.

Before we formally describe the decomposition, we need some terminology. Let $s_1, \dots, s_k$ be the vertices around the outer face with $k = k(n)$ (ordering defined by the outerplanar embedding). For symmetry, we add two dummy vertices s_0, s_{k+1} without neighbours and define $U_1 = \{s_0, s_{k+1}\}$, and $V_1 = \{s_1\}$ (the dummy vertices play no role and can be removed before running Algorithm Embeddings).

The algorithm proceeds in steps with step i creating the (U_i, V_i) pair. For $i > 1$, two vertices s_{j_0}, s_{j_1} with $j_0 < j_1$, define an *interval* at step i if $s_{j_0}, s_{j_1} \in U^{i-1}$, but for $j_0 < l < j_1, s_l \notin U^{i-1}$. If the interval is defined it is the sequence of vertices between s_{j_0}, s_{j_1} (including the endpoints).[9] Let a_i be a median vertex of the vertices in $I \cap V^{i-1}$ (median based on the outerplanar vertex ordering), where I is a step i interval. Define U_i to be the smallest subset of V^{i-1} containing $\{a_i\}$ and also $N_H(N_H(U_i) - V^{i-1}) \cap V^{i-1}$. In other words, U_i is the smallest set of vertices in V^{i-1} including $\{a_i\}$ such that the set of neighbours of U_i excluding the vertices from V^{i-1} (call this set M_i) have the property that the vertices in M_i have no neighbours outside U_i in V^{i-1}. Note that setting U_i as V^{i-1} satisfies the above condition, but it may not be the smallest.

Define $V_i = N_H(U_i) - V^{i-1} = M_i$. We now argue that this is indeed a decomposition. Consider the interval I at the ith step, with s_{j_0}, s_{j_1} as the defining endpoints, and a_i as the median of $I \cap V^{i-1}$.

Lemma 4.5. *Let I be the interval at the ith step. Then $U_i \subseteq I$.*

Proof. U_i can only contain vertices that have a path to a_i but not containing any vertex from U^{i-1} in the path. Since the graph is outerplanar, any path from a_i to any vertex $w \notin I$ passes through either of the endpoints (s_{j_0}, s_{j_1}), both of which are in U^{i-1}. In other

[9] If no interval exists, then all vertices are already part of the decomposition and we are done. Also, there could be more than one interval at each i, in which case we can pick any one.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

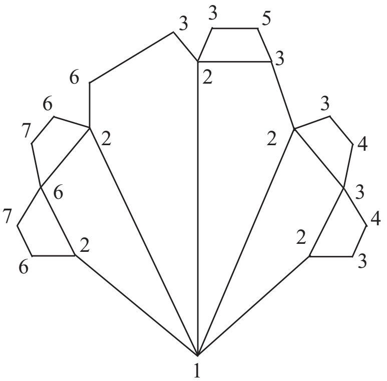

Figure 5. Decomposition of an outer planar graph. Vertices with label i constitute V_i. Neighbours of V_i with lower labels constitute U_i.

words, since the vertices not in I do not have a path to a_i which does not pass through a vertex in U^{i-1}, we have $U_i \subseteq I$. □

Lemma 4.6. *Let I be the interval at the ith step. Then $|U_i| \leqslant |I \cap V^{i-1}| \leqslant 2\triangle$.*

Proof. The first inequality follows as $U_i \subseteq V^{i-1}$ (by construction) and $U_i \subseteq I$ (Lemma 4.5). For the second one we use induction over i. The hypothesis, is true by construction for $i = 1$. Assume the hypothesis holds for $i - 1$. Let J be the interval used by the algorithm at the $(i-1)$th step. By the inductive hypothesis, $|J \cap V^{i-2}| \leqslant 2\triangle$. The interval J is split into several new intervals (at least two as $a_{i-1} \in U_{i-1}$) by the vertices of U_{i-1}, which define the step i intervals. The newly created intervals are of two types: (a) both endpoints are from U_{i-1}, (b) one endpoint is from U_{i-1} and the other is from U_{i-2}. In the intervals of the first type there are at most $2\triangle$ vertices from V_{i-1} (at most $\triangle$ vertices from each of the two endpoints) and no vertex from V^{i-2}. In the intervals of the second type, there are at most $\triangle$ vertices from V_{i-1} adjacent to the endpoint in U_{i-1} and at most $\triangle$ vertices from V^{i-2} (from the inductive hypothesis and the fact that a_{i-1} is the median of $J \cap V^{i-2}$). Therefore, each of the newly created step i intervals (which includes I) have at most $2\triangle$ vertices from V^{i-1}. □

Properties ① and ③ are guaranteed by construction. Let us concentrate on property ②. For a contradiction assume that there exist two vertices v_1 and v_2 in some V_i with the edge (v_1, v_2) in H. Since no triangles exist in H, both v_1 and v_2 should be connected to two different vertices (say, u_1 and u_2) in U_i. However, since the graph is outerplanar, there exists no path from u_1 to u_2 going through any vertices of $U_i \cup V_i$ other than v_1 and v_2. This would mean that we could remove at least one of u_1 or u_2 from U_i without disturbing the condition that it needs to satisfy. This would lead to a contradiction to U_i being the smallest set in V^{i-1} satisfying the condition.

Lemma 4.6 implies that the width of this decomposition is at most $2\triangle \cdot \triangle = 2\triangle^2$ (as $|U_i| \leqslant 2\triangle$). See Figure 5 for an illustration.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

Proposition 4.7. *Let H be a bounded-degree $\mathcal{C}_3$-free outerplanar graph. Then, there exists an ordered bipartite decomposition of H with bounded width.*

4.4. Decomposition of series-parallel graphs

In this section we prove the decomposition property on series-parallel graphs. A series-parallel graph (also called a two-terminal series-parallel graph) is a graph with two distinguished vertices s and t that is obtained as follows. A single edge (s,t) is a series-parallel graph (base case). Let H_a and H_b be two series-parallel graphs with terminals s_a, t_a and s_b, t_b respectively. The graph formed by identifying t_a with s_b is a series-parallel graph with terminals s_a, t_b (series operation is denoted by $\oplus$). The graph formed by identifying s_a with s_b and t_a with t_b is a series-parallel graph with terminals $s_a = s_b$ and $t_a = t_b$ (parallel operation is denoted by $||$).

The algorithm again proceeds in steps with step i creating the (U_i, V_i) pair. In the following, the process of adding a vertex to some V_i is referred to by the term *selecting*. We say a vertex is *finished* once it is added to some U_i, *i.e.*, all its neighbours are selected. The construction is technical, but the basic idea is to first finish the terminals, so that the parallel components separate (for the decomposition purposes). Then, the algorithm finishes some vertex joining two serial components. In both these steps the algorithm might be forced to finish some other vertices too.

To define the decomposition we need more terminology. Let $H = (V_H, E_H)$ be a $[\mathcal{C}_3, \mathcal{C}_5]$-free series-parallel graph with (distinguished) terminals s and t. Let $\mathcal{V}_H = V_{1,H}, V_{2,H}, \dots$ denote a decomposition of H. Let $V_H^i = \bigcup_{j \leqslant i} V_{j,H}$. For a set of vertices S in H, define

$$D_H(i,S) = \{u \in V_H^{i-1} \;:\; \text{there exists } v \in S \text{ such that } (u,v) \in E_H\}.$$

$D_H(i,S)$ represents the set of neighbours of S in H selected in the first $(i-1)$ steps of the algorithm. The algorithm starts by finishing s and t as follows:

$$V_{1,H} = \{s\} \quad \text{and} \quad V_{2,H} = N_H(s) - V_H^1,$$
$$V_{3,H} = \begin{cases} \{t\} \cup N_H(D_H(3,\{t\})) - V_H^2 & \text{if } t \notin V_H^2, \\ \emptyset & \text{otherwise,} \end{cases}$$
$$V_{4,H} = N_H(t) \cup N_H(D_H(4, N_H(t))) - V_H^3.$$

In words, the first four steps of the algorithm achieves: (i) select s, (ii) finish s, (iii) select t unless already selected, (iv) finish t. Define

$$\mathcal{V}_H = V_{1,H}, V_{2,H}, V_{3,H}, V_{4,H}, \mathcal{V}_{H|s,t},$$

where $\mathcal{V}_{H|s,t}$ is defined recursively as follows.

1 *Base case.* If all the vertices in H are selected, $\mathcal{V}_{H|s,t} = \emptyset$.

2 *Parallel case.* If $H = H_a || H_b$, find recursively $\mathcal{V}_{H_a|s,t}$ and $\mathcal{V}_{H_b|s,t}$. Define

$$\mathcal{V}_{H|s,t} = \mathcal{V}_{H_a|s,t}, \mathcal{V}_{H_b|s,t}.$$

3 *Serial case.* If $H = H_a \oplus H_b$, with x as the vertex joining H_a and H_b. Let $s \in V_{H_a}$ and $t \in V_{H_b}$.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

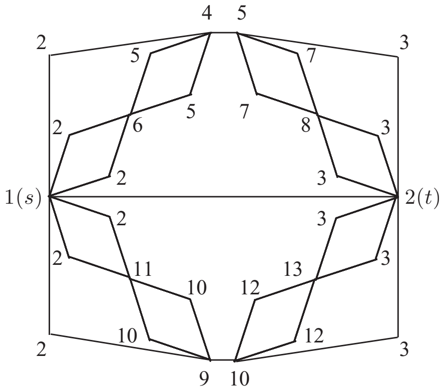

Figure 6. Decomposition of a series-parallel graph. Vertices with label i constitute V_i. Neighbours of V_i with lower labels constitute U_i.

(a) If x is finished, define $\mathcal{V}_{H|s,t} = \mathcal{V}_{H_a|s,x}, \mathcal{V}_{H_b|x,t}$.

(b) If $x \in V_H^4$ (x has already been selected) and x not finished, then finish x. This produces the set

$$V_{5,H} = N_H(x) \cup N_H(D_H(5, N_H(x))) - V_H^4.$$

Define

$$\mathcal{V}_{H|s,t} = V_{5,H}, \mathcal{V}_{H_a|s,x}, \mathcal{V}_{H_b|t,x}.$$

(c) Otherwise, first select x which produces the set

$$V_{5,H} = \{x\} \cup N_H(D_H(5, \{x\})) - V_H^4.$$

Then, finish x. This produces the set

$$V_{6,H} = N_H(x) \cup N_H(D_H(6, N_H(x))) - V_H^5.$$

Define

$$\mathcal{V}_{H|s,t} = V_{5,H}, V_{6,H}, \mathcal{V}_{H_a|s,x}, \mathcal{V}_{H_b|t,x}.$$

The following lemma provides bounds on the sizes of U_i. The proof looks at two possible situations, conditioning on the presence or absence of paths of length 2 or 3 between s and t. Since both $\mathcal{C}_3$ and $\mathcal{C}_5$ are forbidden, it follows that there can either be a path of length 2 or 3 between any two vertices, but not both. This fact will be crucial for implying property ②. See Figure 6 for an example.

Lemma 4.8. *Let H be a $[\mathcal{C}_3, \mathcal{C}_5]$-free series-parallel graph with terminals s and t. Then, the above algorithm finishes $O(\triangle^2)$ vertices in every step (size of all the U_i is $O(\triangle^2)$).*

Proof. The proof is via induction on the size of series-parallel graph. The inductive hypothesis is that if s, t and possibly some vertices in $N_H(s) \cup N_H(N_H(s))$ are the only

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

vertices finished, then the above algorithm finds a decomposition of H by finishing $O(\triangle^2)$ vertices in every step.

The algorithm always finishes s first and then t, and once s and t are finished the parallel components can be handled independently to construct the decomposition. In the process of finishing t, the algorithm could possibly finish some vertices in $N_H(s) \cup N_H(N_H(s))$. Hence, in each of the parallel components H', terminals s, t and possibly some vertices in $N_{H'}(s) \cup N_{H'}(N_{H'}(s))$ are finished. Therefore, inductively a decomposition can be obtained. So the challenging case is when H has just one parallel component. Let $H = H_1 \oplus H_2$ with z as the vertex joining H_1 and H_2. There are three different cases. In each of them the interesting event occurs after s, t, and z are finished, which splits H into H_1 and H_2. Afterwards, decomposition on H_1 and H_2 could be constructed independently.

In the following, we describe the cases under the assumption that there exists no edge between s and t. If there exists such an edge, then the description would remain the same except that the step where t is selected would no longer exist (t is now selected when s is finished). Also, if there is an edge between s and t, then there exists no path of length 2 between s and t, as otherwise there would be a triangle.

Case 1: no path of length 2 or 3 between *s* and *t*. Note that at the step when s is finished no other vertex in H is finished. Later, when t is selected the only vertices in $N_H(s)$ that finish at that step are those which are neighbours of t. This set is $\emptyset$, as otherwise there would be a path of length 2 between s, t. Similarly, at the step when t is finished the only vertices in $N_H(s)$ that finish are those which share a common neighbour with t. This set is also $\emptyset$, as otherwise there would be a path of length 3 between s, t. Now at the step when z is selected some vertices in $N_H(s)$ and $N_H(t)$ could possibly be finished, and at the step when z is finished some vertices in $N_H(s) \cup N_H(t) \cup N_H(N_H(s)) \cup N_H(N_H(t))$ could possibly be finished (this supplies the $O(\triangle^2)$ bound). However, as soon as z is finished, the graphs H_1 and H_2 can be handled independently. Now H_1 is a smaller series-parallel graph with terminals s, z, where s, z and possibly some vertices in $N_{H_1}(s) \cup N_{H_1}(N_{H_1}(s))$ are finished. Therefore, inductively a decomposition of H_1 can be completed. Similarly, H_2 can be viewed as a series-parallel graph with terminals t, z. In H_2, terminals t, z and possibly some vertices in $N_{H_2}(t) \cup N_{H_2}(N_{H_2}(t))$ are finished. Therefore, inductively a decomposition of H_2 can also be completed.

Case 2: paths of length 2 between *s* and *t*. So there is no path of length 3 between s and t. If t has been selected before s is finished, then t is finished together with s (at which step z is also selected). Note that s and t can be finished in the same step because there is no path of length 3. At the step when z is finished some vertices in $N_H(s) \cup N_H(t)$ could possibly be finished. Afterwards, we can invoke induction on both H_1 and H_2. If s is finished before selecting t, then z is finished while selecting t. At the step when z is finished some vertices in $N_H(s)$ could possibly be finished. Later , at the step when t is finished some vertices in $N_H(z)$ could possibly be finished. But again after t is finished, we can invoke induction on both H_1 and H_2. See Figure 7.

Case 3: paths of length 3 between *s* and *t*. So there is no path of length 2 between s and t. There are two subcases based on the distance from s to z.

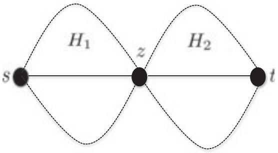

Figure 7. Case 2: the dashed edges may not be present in the graph.

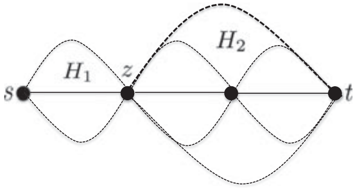

Figure 8. Subcase 3.1.

Subcase 3.1. First assume that the distance between s and z is one. At the step when s is finished z is selected. At the step when t is selected no vertex in H is finished (absence of a path of length 2). At the step when t is finished, z is finished and also some other vertices in $N_H(s)$ could possibly be finished. Hereafter, induction can be invoked over H_1 and H_2. See Figure 8.

Subcase 3.2. Now, if the distance between s and z is two, then the distance between t and z is one. At the step when s is finished no other vertex in H is finished. At the step when t is selected no vertex in H is finished. At the step when t is finished, z gets selected and some vertices in $N_H(s)$ would be finished. Finally, at the step when z is finished some vertices in $N_H(s) \cup N_H(N_H(s)) \cup N_H(t)$ could possibly be finished. Hereafter, induction can be invoked over H_1 and H_2. See Figure 9.

Therefore, a decomposition of H can be obtained with no more than $O(\triangle^2)$ finishing at each step. A more precise upper bound of $2\triangle^2$ can be obtained with a more careful analysis. □

Properties ① and ③ are guaranteed by construction. Property ② follows from the fact that during any step of the above algorithm the set of vertices selected (appearing in the same V_i) is at most distance two (*i.e.*, two neighbourhoods away) from some fixed vertex (see the proof of Lemma 4.8). Since H has no $\mathcal{C}_3$ or $\mathcal{C}_5$, the vertices selected together cannot have any edge between them (*i.e.*, the V_i are independent sets).

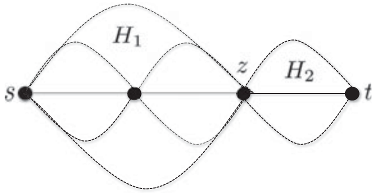

Figure 9. Subcase 3.2.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

The width of this decomposition is $O(\triangle^2)\cdot\triangle = O(\triangle^3)$ as $O(\triangle^2)$ vertices are finished in each step by the above algorithm (Lemma 4.8).

Proposition 4.9. *Let H be a bounded-degree $[\mathcal{C}_3,\mathcal{C}_5]$-free series-parallel graph. Then there exists an ordered bipartite decomposition of H with bounded width.*

4.5. Decomposition of planar graphs

In this section we prove the decomposition property on planar graphs. Define a *thread* as an induced path in H whose vertices are all of degree 2 in H. A k-thread is a thread with k vertices. Let H be a planar graph of girth at least 16. We first prove a structural result on planar graphs.

Lemma 4.10. *Let H be a planar graph of minimum degree* 2 *and girth at least* 16. *Then H always contains a* 3*-thread.*

Proof. Assume without loss of generality that the graph H is connected; otherwise it is sufficient to prove the statement for each of the components. Let $\widehat{H}$ be the graph obtained from H by contracting all degree 2 vertices. Then, $\widehat{H}$ is a planar graph of minimum degree 3.

We first show that $\widehat{H}$ contains a face of degree 5 or less. For contradiction, suppose that all the faces have degree at least 6. Let n be the number of vertices, let m be the number of edges, and let k be the number of faces of $\widehat{H}$. Moreover, let H be the set of faces and V the set of vertices of $\widehat{H}$. Since the degree of each face is at least 6 (where the degree of a face f is the number of edges going around f), $2m = \sum_{f\in F}\deg(f) \geqslant 6k$. Moreover, $2m = \sum_{v\in V}\deg(v) \geqslant 3n$, since the minimum degree in $\widehat{H}$ is at least 3. By Euler's formula[10] and the previous inequalities, we have $m+2 = n+k \leqslant (2m)/3 + m/3 = m$, a contradiction.

Let $\hat{f}$ be a face of H that corresponds to a face of the degree 5 or less in $\widehat{H}$. Since the degree of $\hat{f}$ is at least 16 (the girth is 16), it is easy to see that $\hat{f}$ contains a 3-thread in H. □

In order to define a decomposition, we define a 3-thread partition $X_1,\dots,X_c$ of a planar graph H as a partition of V_H such that each X_i satisfies

$$X_i = \begin{cases} \{a_i\} & \text{if } a_i \text{ is a degree 0 or 1 vertex in the graph induced by } V_H - \bigcup_{j<i} X_j \\ & \quad \textit{on } \text{H}, \\ \{a_i,b_i,c_i\} & \text{if } a_i,b_i,c_i \text{ form a 3-thread in the graph induced by } V_H - \bigcup_{j<i} X_j \text{ on } H. \end{cases}$$

By Lemma 4.10 every planar graph with girth at least 16 has a 3-thread partition. As before, we say that a vertex is selected if we add it to some V_k. Using the 3-thread partition (which can be constructed using Lemma 4.10), a decomposition of a planar graph of girth at least 16 can be constructed by repeating this following simple procedure.

[10] It states that in a planar graph with n vertices, m edges, and k faces, $n-m+k=2$.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

(i) Find the largest index l such that X_l contains a vertex z_l which has not yet been selected but is adjacent to an already selected vertex.
(ii) Define $U_i = N_H(z_l) \cap D^{i-1}$ and $V_i = N_H(U_i) - V^{i-1}$ (where $D^{i-1} = V^{i-1} - U^{i-1}$ as defined earlier).
(iii) Increment i.

Lemma 4.11. *Let H be a planar graph of girth at least* 16. *Then each of the (U_i, V_i) pair created by the above algorithm satisfies $|U_i| \leqslant 2$ and $|V_i| \leqslant 2\triangle$.*

Proof. Let $X_1, \ldots, X_c$ be a 3-thread partition of H. Let $\bar{H}_i$ be the graph induced by $V_H - \bigcup_{j<i} X_j$ on H. The first observation is that a vertex in any X_j $(1 \leqslant j \leqslant c-1)$ has at most one edge connecting it to the vertices in $X_{j+1} \cup \cdots \cup X_c$. Consider some step i of the decomposition (step i is when the (U_i, V_i) pair is created). Let l be the largest index with an unselected vertex z_l. From the previous observation it follows that vertices in $N(z_l)$ that are in $X_1 \cup \cdots \cup X_{l-1}$ are not selected in steps 1 to $i-1$. Assume otherwise. Let u be a vertex belonging to $N(z_l) \cap X_{l'} (l' < l)$ that is selected in the first $i-1$ steps. Then, u needs to have a neighbour in $X_l \cup \cdots \cup X_c - \{z_l\}$, a contradiction since it would imply that u (which is in $X_{l'}$) has two neighbours in $X_l \cup \cdots \cup X_c$. Therefore, until step i none of the neighbours of z_l in $X_1 \cup \cdots \cup X_{l-1}$ have been selected. By definition of threads, z_l could have at most two neighbours in $\bar{H}_i$. The cases where it has two neighbours are as follows: (a) z_l has one neighbour from $X_{l+1} \cup \cdots \cup X_c$ and another from X_l, or (b) z_l has both its neighbours from X_l. This implies that $|N_H(z_l) \cap D^{i-1}| = |U_i| \leqslant 2$, and $|V_i| \leqslant 2\triangle$. □

Properties ① and ③ of the decomposition are again guaranteed by construction. Property ② is satisfied because $|U_i| \leqslant 2$ and the vertices in U_i are neighbours of z_l, so the vertices in V_i cannot have edges between them, otherwise it will result in a cycle of length 5. Since this holds for every V_i, the V_i are independent sets.

The width of this decomposition is at most $2\triangle$ (as $|U_i| \leqslant 2$ from Lemma 4.11).

Proposition 4.12. *Let H be a bounded-degree planar graph of girth at least* 16. *Then, there exists an ordered bipartite decomposition of H with bounded width.*

5. Negative result for ordered bipartite decomposition

As mentioned earlier, only graphs of bounded degree have a chance of having a decomposition of bounded width. So a natural question to ask is whether all bounded-degree graphs with a decomposition have one of bounded width. In this section, we answer this question negatively by showing that every unbounded-width grid graph (remember that the width of an $n_1 \times n_2$ grid graph is $\min\{n_1, n_2\}$) fails to satisfy this condition. For simplicity, we will only consider $\sqrt{n} \times \sqrt{n}$ grid graphs, but our proof techniques extend to other cases as well.

Let $H = (V_H, E_H)$ be a $\sqrt{n} \times \sqrt{n}$ grid graph with $V_H = \{(i,j) : 0 \leqslant i, j \leqslant \sqrt{n} - 1\}$ and $E_H = \{((i,j),(i',j')) : i = i' \text{ and } |j - j'| = 1 \text{ or } |i - i'| = 1 \text{ and } j = j'\}$. We now show that

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

any decomposition of H has a width of at least $\Omega(\sqrt{n})$. Let $V_1,\ldots,V_\ell$ be any decomposition of H. Consider any 2×2 square of H defined by vertices a,c,b,d (in clockwise order). Assume without loss of generality that the vertex c has the smallest label (given by the decomposition) among vertices a,b,c, and d, and let the label on c be l. The two neighbours a,b of the vertex c always have the same label $l' > l$. The fourth vertex d has any label l'' with $l'' \geqslant l$ and $l'' \neq l'$. We define a new graph $H' = (V_H, E_{H'})$ on the same set of vertices by putting the edge (a,b) into $E_{H'}$. Note that all vertices in a connected component in H' have the same label, and thus need to be chosen together in the decomposition (*i.e.*, all vertices in a connected component in H' appear in the same V_k in the decomposition).

Let $\mathcal{H}_D$ be a class of graphs on vertex set V_H with exactly one diagonal in every 2×2 square (and no other edges). That is, any graph $H_D = (V_H, E_D)$ from $\mathcal{H}_D$ has, for every (i,j) with $0 \leqslant i,j \leqslant \sqrt{n}-2$, exactly one of the edges $((i,j),(i+1,j+1)),((i,j+1),(i+1,j))$ in E_D and no other edges are in E_D. Note that $H' \in \mathcal{H}_D$. The following theorem shows that any graph $H_D \in \mathcal{H}_D$ has the property that there is a connected component touching the top and bottom or left and right (and therefore $H' \in \mathcal{H}_D$ also has this property). Note that (as mentioned before) every connected component in $H' \in \mathcal{H}_D$ would have to be chosen together in the decomposition, implying that the width of the decomposition is $\Omega(\sqrt{n})$.

Theorem 5.1. *Consider any graph $H_D \in \mathcal{H}_D$. There exists a connected component of H_D that contains at least one vertex from every row or at least one vertex from every column in the grid graph.*

Proof. Assume H_D does not have a connected component that contains a vertex of every row. Let $H_U = (V_U, E_U)$ be the subgraph of H_D generated by all the vertices connected to the top row, *i.e.*, H_U is a collection of those connected components in H_D that have at least one vertex from the top row. By assumption, H_U does not contain any vertices from the bottom row.

For every 2×2 subgrid with vertices a,c,b,d and edge $(a,b)\in E_{H'}$, we call (a,b) a *boundary edge* if exactly one of c,d is in V_U and neither of a or b are in V_U. Let $H_B = (V - V_U, E_B)$ be the subgraph of H_D where E_B is the set of boundary edges. We assign the colour red to all the vertices in V_U and black to all the vertices in $V - V_U$. In the following two claims we make some observations about the structure of H_B. For a vertex v, let $C(v)$ indicate whether the vertex is coloured red (r) or black (b).

Claim 5.2. *There are no degree* 3 *vertices in H_B, i.e., all vertices in H_B have degree* 0, 1, 2 *or* 4.

Proof. Assume the contrary. Let u be a degree 3 black vertex. Let $(u,v_1),(u,v_2),(u,v_3)$ be the only edges incident on u in H_B.

By choice, $C(u) = b, C(v_1) = b, C(v_2) = b, C(v_3) = b$. For the other vertices, there are only two possibilities:

(i) $C(v_4) = b, C(w_1) = b, C(w_2) = r, C(w_3) = b, C(w_4) = r$, and

(ii) $C(v_4) = b, C(w_1) = r, C(w_2) = b, C(w_3) = r, C(w_4) = b$.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

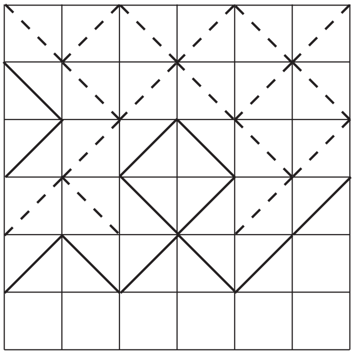

Figure 10. The negative result. The dashed diagonal lines are the edges in H_U, and the solid diagonal lines are the edges in H_B. There exists a component in H_B that spans from the left to right boundary.

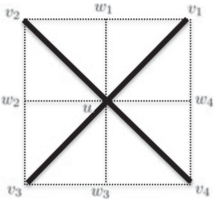

Figure 11. The solid lines are the edges in H_B, whereas the dotted lines are at the edges in the grid. If $C(u) = b, C(v_1) = b, C(v_2) = b, C(v_3) = b$, then the two possibilities for colour assignments to other vertices are (i) $C(v_4) = b, C(w_1) = b, C(w_2) = r, C(w_3) = b, C(w_4) = r$ and (ii) $C(v_4) = b, C(w_1) = r, C(w_2) = b, C(w_3) = r, C(w_4) = b$.

Since all the edges in H_D are all either between two red vertices or two black vertices and every 2×2 subgrid has exactly one edge, v_4 is black and there exists an edge between (u, v_4) in H_B. Therefore, every vertex in H_B has degree either $0, 1, 2,$ or 4. See Figure 11. □

As in the previous claim, by considering all possibilities for the neighbours of u being in V_U or not, one can conclude immediately that all vertices of degree 1 are on the left or right border and there are odd numbers of degree 1 vertices on each border.

Claim 5.3. *All the degree* 1 *vertices of* H_B *are on either the left or right border of the grid graph* H. *Moreover, there is an odd number of degree* 1 *vertices of* H_B *on the left and right borders.*

Every connected component in H_B has an even number of degree 1 vertices. From Claim 5.3, we know that degree 1 vertices only occur at the left and right boundaries of H_B, and there are odd numbers of them on the two boundaries. Putting these two statements together implies that there exists a component in H_B (therefore in H_D) that connects the left and the right border. This finishes the proof of Theorem 5.1. See Figure 10 for an illustration. □

Corollary 5.4 (negative result). *Every decomposition of a $\sqrt{n} \times \sqrt{n}$ grid graph H has a width of $\Omega(\sqrt{n})$.*

6. Conclusions and open problems

The natural question arising from this work is what other classes of graphs have an ordered bipartite decomposition and, more importantly, which of them have one of bounded-width decomposition. Other than the graph classes mentioned above, the bounded-degree $[\mathcal{C}_3, \mathcal{C}_5]$-free Halin graphs [18], where degree 2 vertices are allowed, and the hexagonal grid graphs are some other interesting graph classes which have bounded-width decompositions. Most of the graph classes we have considered appear to have small treewidth. So a natural question would be to relate these two decomposition schemes. However, we show in Appendix B that the treewidth and the width of an ordered bipartite decomposition are incomparable.

Another interesting problem would be to investigate the general complexity of the ordered bipartite decomposition and possibly characterize its relation to other existing graph decomposition schemas. Bounded-width decomposition is a natural sufficient condition for the class of algorithms based on the principle of Algorithm Count to give an FPRAS almost always. But the necessary condition for the general approach to work is still unclear. Finally, a challenging open problem is to obtain any such general result for counting in arbitrary dense graphs.

Acknowledgements

We thank Andrzej Ruciński for pointing us to [33] and Piotr Berman for simplifying the proofs in Section 5. The authors would also like to thank Sofya Raskhodnikova, Adam Smith, and Martin Tancer for helpful comments and discussions.

We also thank anonymous referees for pointing out an error in Definition 3.1 in an earlier version of this paper.

Appendix A: Extension to the disjoint triangle case

For simplicity we will discuss only the case where H is a union of $n/3$ vertex-disjoint triangles (other cases where H is a union of fewer vertex-disjoint triangles can be handled similarly). Even though H does not have a decomposition, there is a simple FPRAS for counting copies of H in random graphs. Let $s_1, \ldots, s_n$ be the vertices in H, with every triplet $s_{3i+1}, s_{3i+2}, s_{3i+3}$ forming a triangle in H (for $i = 0, .., n/3 - 1$).

Let $Z = X/\operatorname{aut}(H)$ be the output of Algorithm Embeddings for inputs H and $G \in \mathcal{G}(n, p = \text{constant})$, but where each $V_i = \{s_i\}$ and $\ell = n$ (even though $V_1, \ldots, V_\ell$ is not an ordered bipartite decomposition). As in Proposition 3.7, we will again investigate the ratio

$$\frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X^2]]}{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]^2},$$

which equals the critical ratio of averages.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

The numerator,

$$\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X^2]] = \mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_1^2 X_2^2 \cdot \cdots \cdot X_n^2]] = \mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_1^2 X_2^2 X_3^2]] \cdot \cdots \cdot \mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_{n-2}^2 X_{n-1}^2 X_n^2]].$$

The last equality follows because after embedding each triangle, the subgraph of G into which nothing has yet been embedded is random, with the original edge probability p. Consider a representative term from this product,

$$\begin{aligned}\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_{3i+1}^2 X_{3i+2}^2 X_{3i+3}^2]] &= \mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_{3i+1}^2]]\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_{3i+2}^2 X_{3i+3}^2]] \\ &= (n-3i)^2 \mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_{3i+2}^2 X_{3i+3}^2]].\end{aligned}$$

Here, as earlier, we relied on the fact that the graph into which we embed the vertex s_{3i+2} is random. Let $m = X_{3i+2}$ and $m' = X_{3i+3}$. Therefore, m denotes the number of ways of embedding the vertex s_{3i+2} and m' denotes the number of ways of embedding the vertex s_{3i+3}. Since the number of edges incident on the vertices in G is binomially distributed,

$$\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_{3i+1}^2 X_{3i+2}^2 X_{3i+3}^2]]$$

equals

$$(n-3i)^2 \sum_{m=0}^{n-3i-1} m^2 \left(\sum_{m'=0}^{m-1} m'^2 \binom{m-1}{m'} p^{m'}(1-p)^{m-1-m'}\right) \times \binom{n-3i-1}{m} p^m (1-p)^{n-3i-1-m}.$$

Let L_i denote the number of embeddings of a triangle in a random graph from $\mathcal{G}(n-3i, p)$. Then, the denominator

$$\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]^2 = \mathbb{E}[L_0]^2 \cdot \cdots \cdot \mathbb{E}[L_{n/3-1}]^2.$$

Note that $\mathbb{E}[L_i] = \binom{n-3i}{3} 3!p^3$. Using the above equalities, the critical ratio of averages can be bounded to

$$\frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X^2]]}{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X]]^2} = \prod_{i=0}^{n/3-1} \frac{\mathbb{E}_{\mathcal{G}}[\mathbb{E}_{\mathcal{A}}[X_{3i+1}^2 X_{3i+2}^2 X_{3i+3}^2]]}{\mathbb{E}[L_i]^2} \leqslant \prod_{i=0}^{n/3-1} \left(1 + \frac{c}{n-3i-4}\right),$$

for a constant c. Again, we obtain a polynomial bound on the critical ratio of averages, which translates to an FPRAS for counting copies of H in $G \in \mathcal{G}(n, p = \text{constant})$.

Appendix B: Ordered bipartite decomposition versus treewidth

In this section we show that the treewidth and the width of an ordered bipartite decomposition are incomparable. In one direction, consider a star graph. The treewidth is 1, but no ordered bipartite decomposition of width less than $n/2$ exists. For the other direction, we consider the 1-subdivision graph of a constant-degree expander, as explained below.

Let H be a constant-degree expander graph. Consider the 1-subdivision graph $S(H)$ of H. From Proposition 4.4, $S(H)$ has an ordered bipartite decomposition of bounded width. So the only fact that remains to be verified is that the vertex expansion ratio of $S(H)$ is a constant.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

Lemma B.1. *A* 1*-subdivision graph of a constant-degree expander is an expander.*

Proof. Let A be a set of vertices in H. Let α (= constant) denote the vertex expansion ratio of H and $\triangle$ denote the maximum degree in H. Let $S(H)$ denote the 1-subdivision graph of H. Let B be a subset of vertices from $N_{S(H)}(A)$. We consider the vertex expansion ratios for two different scenarios of B.

- *Case* $\mathbf{B} = \emptyset$. In this case, $|N_{S(H)}(A)| \geqslant |N_H(U)| \geqslant \alpha|A|$.
- *Case* $\mathbf{B} \neq \emptyset$. First assume that $B = N_{S(H)}(A)$. Under this assumption, $N_{S(H)}(A \cup B) = N_H(A)$. Say that $|N_H(A)| = \lambda$. Now, even if $B \subset N_{S(H)}(A)$, $|N_{S(H)}(A \cup B)|$ is at least λ. Therefore,
$$|N_{S(H)}(A \cup B)| = \lambda \geqslant \alpha|A| \geqslant \frac{\alpha}{\triangle + 1}(|A| + |B|).$$

The final case to consider involves a set of vertices C in $S(H)$, which are not in H. In this case, $|N_{S(H)}(C)| \geqslant |C|/\triangle$.

From the above case analysis it is clear that the vertex expansion ratio of $S(H)$ is a constant, and the proof follows. □

$S(H)$ has constant expansion which implies a treewidth $\Theta(n)$ [20], whereas $S(H)$ has an ordered bipartite decomposition of bounded width. Therefore, treewidth and the width of an ordered bipartite decomposition are incomparable.

References

[1] Alon, N., Frieze, A. M., and Welsh, D. (1995) Polynomial time randomized approximation schemes for Tutte–Gröthendieck invariants: The dense case. *Random Struct. Alg.* **6** 459–478.

[2] Alon, N., Yuster, R. and Zwick, U. (1995) Color-coding. *J. Assoc. Comput. Mach.* **42** 844–856.

[3] Artymiuk, P. J., Bath, P. A., Grindley, H. M., Pepperrell, C. A., Poirrette, A. R., Rice, D. W., Thorner, D. A., Wild, D. J., Willett, P., Allen, F. H., and Taylor, R. (1992) Similarity searching in databases of three-dimensional molecules and macromolecules. *J. Chem. Inform. Comput. Sci.* **32** 617–630.

[4] Arvind, V., and Raman, V. (2002) Approximation algorithms for some parameterized counting problems. In *Algorithms and Computation*, Vol. 2518 of *Lecture Notes in Computer Science*, Springer, pp. 453–464.

[5] Bezáková, I., Bhatnagar, N., and Vigoda, E. (2007) Sampling binary contingency tables with a greedy start. *Random Struct. Alg.* **30** 168–205.

[6] Chien, S. (2004) A determinant-based algorithm for counting perfect matchings in a general graph. In *Proc. 15th Annual ACM–SIAM Symposium on Discrete Algorithms*, pp. 728–735.

[7] Diestel, R. (2000) *Graph Theory*, second edition, Springer.

[8] Dong, H., Wu, Y., and Ding, X. (1988) An ARG representation for Chinese characters and a radical extraction based on the representation. In *International Conference on Pattern Recognition*, IEEE, pp. 920–922.

[9] Dyer, M., Frieze, A. M., and Jerrum, M. (1998) Approximately counting Hamilton paths and cycles in dense graphs. *SIAM J. Comput.* **27** 1262–1272.

[10] Erdős, P., and Rényi, A. (1960) On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* **5** 17–61.

[11] Flum, J., and Grohe, M. (2004) The parameterized complexity of counting problems. *SIAM J. Comput.* **33** 892–922.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press

[12] Frieze, A. M., and Jerrum, M. (1995) An analysis of a Monte Carlo algorithm for estimating the permanent. *Combinatorica* **15** 67–83.

[13] Frieze, A. M., Jerrum, M., Molloy, M. K., Robinson, R., and Wormald, N. C. (1996) Generating and counting Hamilton cycles in random regular graphs. *J. Algorithms* **21** 176–198.

[14] Frieze, A. M., and McDiarmid, C. (1997) Algorithmic theory of random graphs. *Random Struct. Alg.* **10** 5–42.

[15] Frieze, A. M., and Suen, S. (1992) Counting the number of Hamilton cycles in random digraphs. *Random Struct. Alg.* **3** 235–242.

[16] Fürer, M., and Kasiviswanathan, S. P. (2004) An almost linear time approximation algorithm for the permanent of a random (0–1) matrix. In *Foundations of Software Technology and Theoretical Computer Science: FSTTCS 2004*, Vol. 3328 of *Lecture Notes in Computer Science*, Springer, pp. 263–274.

[17] Fürer, M., and Kasiviswanathan, S. P. (2005) Approximately counting perfect matchings in general graphs. In *ALENEX/ANALCO*, SIAM, pp. 263–272.

[18] Halin, R. (1971) Studies on minimally n-connected graphs. In *Combinatorial Mathematics and its Applications* (D. J. A. Welsh, ed.), Academic Press, pp. 129–136.

[19] Hammersley, J. M. (1966) Existence theorems and Monte Carlo methods for the monomer–dimer problem. *Research Papers in Statistics*, Wiley, London, 125–146.

[20] Heun, V., and Mayr, E. W. (2002) Embedding graphs with bounded treewidth into optimal hypercubes. *J. Algorithms* **43** 17–50.

[21] Janson, S., Łuczak, T., and Ruciński, A. (2000) *Random Graphs*, Wiley-Interscience.

[22] Jerrum, M. (2003) *Counting, Sampling and Integrating: Algorithms and Complexity*, Birkhäuser.

[23] Jerrum, M., and Sinclair, A. (1989) Approximating the permanent. *SIAM J. Comput.* **18** 1149–1178.

[24] Jerrum, M., and Sinclair, A. (1993) Polynomial-time approximation algorithms for the Ising model. *SIAM J. Comput.* **22** 1087–1116.

[25] Jerrum, M., Sinclair, A., and Vigoda, E. (2004) A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. Assoc. Comput. Mach.* **51** 671–697.

[26] Jerrum, M., Valiant, L., and Vazirani, V. (1986) Random generation of combinatorial structures from a uniform distribution. *Theoret. Comput. Sci.* **43** 169–188.

[27] Kasteleyn, P. W. (1967) *Graph Theory and Crystal Physics* (F. Harary, ed.), Academic Press.

[28] Knuth, D. E. (1975) Estimating the efficiency of backtrack programs. *Math. Comp.* **29** 121–136.

[29] Levinson, R. (1992) Pattern associativity and the retrieval of semantic networks. *Comput. Math. Appl.* **23** 573–600.

[30] Luks, E. M. (1982) Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.* **25** 42–65.

[31] Rasmussen, L. E. (1994) Approximating the permanent: A simple approach. *Random Struct. Alg.* **5** 349–362.

[32] Rasmussen, L. E. (1997) Approximately counting cliques. *Random Struct. Alg.* **11** 395–411.

[33] Riordan, O. (2000) Spanning subgraphs of random graphs. *Combin. Probab. Comput.* **9** 125–148.

[34] Sankowski, P. (2003) Alternative algorithms for counting all matchings in graphs. In *STACS*, Vol. 2607 of *Lecture Notes in Computer Science Volume*, Springer, pp. 427–438.

[35] Stahs, T., and Wahl, F. M. (1992) Recognition of polyhedral objects under perspective views. *Comput. Artificial Intelligence* **11** 155–172.

[36] Toda, S. (1989) On the computational power of PP and $\oplus P$. In *FOCS*, IEEE, pp. 514–519.

[37] Valiant, L. G. (1979) The complexity of computing the permanent. *Theoret. Comput. Sci.* **8** 189–201.

https://doi.org/10.1017/S0963548314000339 Published online by Cambridge University Press