Approximately Counting Embeddings into Random Graphs

Martin Fürer* and Shiva Prasad Kasiviswanathan

Computer Science and Engineering, Pennsylvania State University
{furer,kasivisw}@cse.psu.edu

Abstract. 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 (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 new decomposition is a labeling of the vertices generating a sequence of bipartite graphs. The decomposition permits us to break the problem of counting embeddings of large subgraphs into that of counting embeddings of small subgraphs. Using this, 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, whereas unbounded-width grid graphs are excluded.

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 [1,2,3,4]. 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 [5].

* Supported in part by NSF award CCF-0728921.

A. Goel et al. (Eds.): APPROX and RANDOM 2008, LNCS 5171, pp. 416–429, 2008.

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 [6]). 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 [7] 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 [8] and positive results for some special cases of approximate counting [9]. 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. Two key examples are spanning trees in a graph, and perfect matchings in a planar graph [10]. A few more problems such as counting perfect matchings in a bipartite graph (a.k.a. (0-1) permanent) [11], counting all matchings in a graph [12], counting labeled subgraphs of a given degree sequence in a bipartite graph [13], counting combinatorial quantities encoded by the Tutte polynomial in a dense graph [14], and counting Hamilton cycles in dense graphs [15], can be done approximately. But problems like 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) [16], 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: 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.

Previous attempts at solving these kinds of problems have not been very fruitful. For example, even seemingly simple problems like 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 [17]). In this paper we present new techniques that can not only handle simple graphs like cycles, but also major subclasses of more complicated graph classes like outerplanar, series-parallel, planar etc.

The theory of random graphs was initiated by Erdős and Rényi [18]. The most commonly used models of random graphs are $\mathcal{G}(n, p)$ and $\mathcal{G}(n, m)$. Both models specify a distribution on n-vertex graphs. In $\mathcal{G}(n, p)$ each of the $\binom{n}{2}$ edges is added to the graph independently with probability p and $\mathcal{G}(n, m)$ 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 analyzing 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 [17] for more). One of the most well-studied problem is that of counting perfect matchings in graphs. For this problem, Jerrum and Sinclair [19] have presented a simulation of a Markov chain that almost always is an fpras (extended to all bipartite graphs in [11]). Similar results using other approaches were obtained later in [20,21,22,23]. Another well-studied problem is that of counting Hamiltonian cycles in random digraphs. For this problem, Frieze and Suen [24] have obtained an fpras, and later Rasmussen [21] has presented a simpler fpras. Afterwards, Frieze *et al.* [25] have obtained similar results in random regular graphs. Randomized approximation schemes are also available for counting the number of cliques in a random graph [26]. However, there are no general results for counting copies of an arbitrary given 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 which 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 labeling of vertices such that every edge is between vertices with different labels and for every vertex all neighbors with a higher label have identical labels. The labeling 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 decomposition, while at the same time many simple small graphs (like a triangle) 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, the function Count (Fig. 1) 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. *The Function Count 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, $\mathbb{E}[Z] = \sum_i p_i \cdot \frac{1}{p_i} = |\mathcal{S}|$.

Function Count: `Assume` $p_i > 0$ `for all` i `and` $\sum_{i=1}^{z} p_i \leq 1$
`If some element` x_i `is picked by` $\mathcal{A}$ `then output` $Z = \frac{1}{p_i}$
`Else output` $Z = 0$

Fig. 1. Unbiased estimator for the cardinality of $\mathcal{S}$

Similar schemes of counting have previously been used by Hammersley [27] and Knuth [28] in other settings. Recently, this scheme has been used by Rasmussen for approximating the permanent of a (0-1) matrix [21], and later for approximately counting cliques in a graph [26]. 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 [29,23]. This is however 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 [30].

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 interesting question 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 subsequence 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 $\bar{H}_1$ in G, then randomly an embedding of $\bar{H}_2$ in 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 the function 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 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 later 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. In this extended abstract, we show that graphs of degree at most two, bounded-degree forests, bounded-width grid (lattice) graphs, subdivision of bounded-degree graphs, bounded-degree outerplanar graphs of girth at least four, and bounded-degree $[\mathcal{C}_3, \mathcal{C}_5]$-free series-parallel graphs, 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 and 4).

Theorem 1 (Main Result). *Let H be a simple graph where each connected component is one of the following: graph of degree at most two, bounded-degree tree, bounded-width grid, subdivision of a bounded-degree graph, bounded-degree $\mathcal{C}_3$-free outerplanar graph, bounded-degree $[\mathcal{C}_3, \mathcal{C}_5]$-free series-parallel graph, or planar graphs of girth at least 16. Then, for almost all graphs G, there exists an fpras for estimating the number of copies of H in G.*

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 [20,21,22,24]. 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 [17]). We omit further discussion of this problem.

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

2 Definitions and Notation

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,

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

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 [31].

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 $aut(H)$ to denote the size of its automorphism group. For a bounded-degree graph H, $aut(H)$ can be evaluated in polynomial time [32]. Most of the other graph-theoretic concepts which we use, such as planarity are covered in standard text books (see, e.g., [33]).

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 and E_F to denote its vertex set and edge set, respectively. 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 neighborhood of S in F. $F[S]$ denotes the subgraph of F induced by S. We use $C_H(G)$ to denote the number of copies of H in G. Let $L_H(G) = C_H(G) \cdot aut(H)$ denote the number of embeddings (or labeled copies) of H in G. For a random graph G, we will be interested in quantities $\mathbb{E}[C_H(G)^2]$ and $\mathbb{E}[C_H(G)]^2$.

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 terms 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 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.

Because of space constraints proofs are omitted from the following presentation; all omitted proofs can be found in [34].

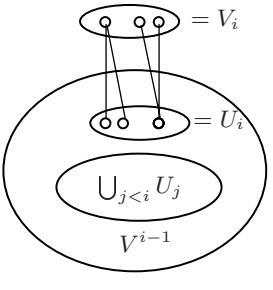
3 Approximation Scheme for Counting Copies

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.

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 $\forall i\, \exists j$ 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 neighbor of a vertex $v \in V_i$ is in some V_j ($j > i$), then all other neighbors 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 $V^i = \bigcup_{j \le i} V_j$. Define $U_i = N_H(V_i) \cap V^{i-1}$. U_i is the set of neighbors 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. The *width* of an ordered bipartite decomposition is the size (number of edges) of the largest H_i.

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

Lemma 1. *Let $V_1, \ldots, V_\ell$ be a decomposition of a graph $H = (V_H, E_H)$. Then, the following assertions are true. (i) Each of the U_i is an independent set in H (H_i is a bipartite graph). (ii) The edge set E_H is partitioned into $E_{H_1}, \ldots, E_{H_\ell}$.*

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 though 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). For us the parameter ℓ plays no role.

Algorithm Embeddings(G,H)

Initialize $X \leftarrow 1$, $Mark(0) \leftarrow \emptyset$, $\varphi(\emptyset) \leftarrow \emptyset$
let $V_1, \ldots, V_\ell$ denote an ordered bipartite decomposition of H
for $i \leftarrow 1$ to ℓ do
  let $G_f \leftarrow G[V_G - Mark(i-1) \cup \varphi(U_i)]$
  compute X_i (the number of embeddings of H_i in G_f with U_i mapped by φ)
  pick an embedding u.a.r. (if one exists) and use it to update φ
  if no embedding exists, then set Z to 0 and terminate
  $X \leftarrow X \cdot X_i$
  $Mark(i) \leftarrow Mark(i-1) \cup \varphi(V_i)$
$Z \leftarrow X/aut(H)$
output Z

The input to the algorithm Embeddings is the template graph H together with its decomposition and the base graph G. The algorithm tries to construct a bijection φ between the vertices of H and G. V_i represents the set of vertices of H which get embedded into G during the i^{th}-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 $X > 0$, then the function φ represents an embedding of H in G (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), and Z is an unbiased estimator for the number of copies of H in G.

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

3.1 FPRAS for Counting in Random Graphs

Since the 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 $C_H(G)$. From here on, we abbreviate $C_H(G)$ as C. The ratio $\mathbb{E}_{\mathcal{A}}[Z^2]/\mathbb{E}_{\mathcal{A}}[Z]^2$ is commonly referred to as the *critical ratio*.

We now concentrate on showing that for random graphs the algorithm is an fpras. A few of the technical details of our proof are somewhat similar to previous applications of this sampling idea, such as that for counting perfect matchings [21,23]. The simpler techniques in these previous results, however, are limited to handling one edge per stage (therefore, working only when H is a matching). Our algorithm embeds a small sized subgraph at every stage. The key for 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 [21,23]). Adding this to the fact that we can do a stage-by-stage analysis of the critical ratio (thanks to the decomposition property which ensures the graph stays essentially random), provides the ingredients for the fpras.

The analysis will be done for a worst-case graph H under the assumption that the sizes of the bipartite graphs H_i's are bounded by a universal constant w, and a random graph G. 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 are closely related. For this purpose, we take a detour through the $\mathcal{G}(n,m)$ model. The ratio $\mathbb{E}[C^2]/\mathbb{E}[C]^2$ plays an important role here and for bounding it we use a recent result of Riordan [35]. The result (stated below) studies the related question of when a random graph G is likely to have a spanning subgraph isomorphic to H.

In the following, N is used to denote $\binom{n}{2}$. We say an event holds with high probability (w.h.p.), if it holds with probability tending to 1 as $n \to \infty$.

Theorem 2. *(Riordan [35]) 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: $\alpha N \geq n$, and $pN, (1-p)\sqrt{n}, np^{\gamma}/\triangle^4 \to \infty$, where*

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

Then, w.h.p. a random graph $G \in \mathcal{G}(n,pN)$ has a spanning subgraph isomorphic to H.

The quantity γ is closely related to twice the maximum average degree of a subgraph of H. The idea behind the proof is to use Markov's inequality to bound $\Pr[C = 0]$ in terms of $\mathbb{E}[C]$ and $Var[C]$. The main thrust lies in proving that $\mathbb{E}[C^2]/\mathbb{E}[C]^2 = 1 + o(1)$. Now by just following Riordan's proof, we obtain the following result.

Proposition 2. *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. Let $\nu = \max\{2, \gamma\}$. Suppose that the following conditions hold: $pN, np^{\nu}/\triangle^4 \to \infty$ and $\alpha^3 N p^{-2} \to 0$. Then, w.h.p. a random graph $G \in \mathcal{G}(n, pN)$ satisfies $\mathbb{E}[C^2]/\mathbb{E}[C]^2 = 1 + o(1)$. In particular, if H is a bounded-degree graph on n vertices. Then, w.h.p. a random graph $G \in \mathcal{G}(n, \Omega(n^2))$ satisfies $\mathbb{E}[C^2]/\mathbb{E}[C]^2 = 1 + o(1)$.*

Note that some of the conditions in Proposition 2 are rephrased from Theorem 2. These are the conditions in the proof of Theorem 2 that are needed for bounding $\mathbb{E}[C^2]/\mathbb{E}[C]^2$. We will be interested in bounded-degree graphs H. For a bounded-degree graph H, both $\triangle$ and γ are constants. Additionally, we will be interested in dense random graphs (where the conditions of Proposition 2 are satisfied). Interpreting Proposition 2 in the $\mathcal{G}(n,p)$ model by using known results for asymptotic equivalence between $\mathcal{G}(n,m)$ and $\mathcal{G}(n,p)$ models (e.g., see Proposition 1.12 of [36]) yields

Lemma 2. *Let H be a bounded-degree 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 $C \geq \mathbb{E}[C]/\omega$.*

Remark: Since C is fairly tightly concentrated around its mean, a rudimentary approximation for C is just $\mathbb{E}[C] = \frac{n! p^{e_H}}{aut(H)}$ (as $v_H = n$). However, this naive approach doesn't produce for *any* $\epsilon > 0$, an $(1 \pm \epsilon)$-approximation for C (see, e.g., [21,20,24,26]).

Using the above result we investigate the performance of algorithm Embeddings when G is a random graph. In this extended abstract, we don't try to optimize the order of the polynomial arising in the running time analysis. Even though for simple template instances such as matchings or cycles, one could easily determine the exact order. The proof idea is to break the critical ratio analysis of the large subgraph into a more manageable critical ratio analysis of small subgraphs.

Theorem 3 (Main Theorem). *Let H be a n-vertex 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.*

Summarizing: if H has a decomposition of bounded width w, then for almost all graphs G, running the algorithm Embeddings $\mathrm{poly}(n)\epsilon^{-2}$ times and taking the mean, 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 the algorithm also takes polynomial time (as H has bounded width), we get an fpras.

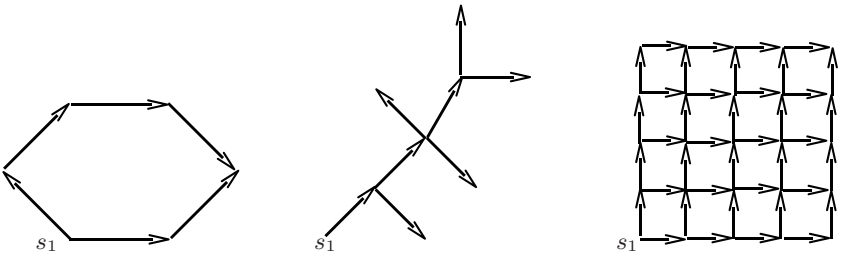


Fig. 2. Decomposition of a cycle, tree, and grid. The graphs are actually undirected. The arrows just connect the vertices of U_i to their neighbors in V_i. All out-neighbors of a vertex are in the same V_i, and all in-neighbors of a vertex are in the same U_i.

4 Graphs with Ordered Bipartite Decomposition

We divide this section into subsections based on the increasing complexity of the graph classes. Some of the later graph classes include the ones that will be covered earlier. We will prove the following result in this section.

Theorem 4. *Let H be a graph where each connected component is one of the following: graph of degree at most two, tree, bounded-width grid, subdivision graph, $\mathcal{C}_3$-free outerplanar graph, $[\mathcal{C}_3, \mathcal{C}_5]$-free series-parallel graph, or planar graph 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.*

Decompositions of subdivision graphs, $[\mathcal{C}_3, \mathcal{C}_5]$-free series-parallel graphs are omitted in this extended abstract (see [34]). From now onwards, we concentrate on connected components of the graph H. If H is disconnected a decomposition is obtained by combining the decomposition of all the connected components (in any order). We will abuse notation and let H stand for both the graph and a connected component in it. $\triangle$ is the maximum degree in H. For constructing the decomposition, the following definitions are useful, $U^i = \bigcup_{j \le i} U_j, V^i = \bigcup_{j \le i} V_j,$ and $D^i = V^i - U^i$.

4.1 Some Simple Graph Classes

We start off by considering simple graph classes such as graphs of degree at most two (paths and cycles), trees, and grid graphs. Fig. 2 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 $1 \le i \le k+1$.
- **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 $3 \le i \le k-1$. Cycles of length three (triangles) don't have a decomposition, but can easily be handled separately (see [34]). Actually, if $H = H_1 \cup H_2$, where graphs

H_1 and H_2 are disjoint, H_1 has a decomposition of bounded width, and H_2 consists of a vertex disjoint union of triangles, then again, there exists an fpras for estimating C. This also completes the claim for graphs of degree at most two in Theorem 1.

- **Trees:** For a tree H, $V_1 = \{s_1\}$, where s_1 is any vertex in H. For $i \geq 2$, let U_i be any vertex from D^{i-1}, then V_i is the set of neighbors 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 the decomposition is at most $\triangle$.
- **Grids:** Let w_0 be the width of the grid graph H. 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 from s_1, the sizes of the V_i's are bounded if w_0 is bounded. Consequently, the width of the decomposition is bounded if w_0 is bounded. This construction also extends to higher dimensional grid graphs.

4.2 Outerplanar Graphs

A graph is outerplanar if it has a planar embedding such that all vertices are on the same face. Let H be a $\mathcal{C}_3$-free outerplanar graph. The idea behind the decomposition is that vertices in U_i partitions 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 $s_1, \ldots, s_k$ 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 neighbors 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 the algorithm Embeddings). For $i \geq 1$, two vertices s_{j_0}, s_{j_1} with $j_0 < j_1$, define a stage i interval if $s_{j_0}, s_{j_1} \in U^i$, but for $j_0 < l < j_1, s_l \notin U^i$. If the interval is defined it is the sequence of vertices between s_{j_0}, s_{j_1} (including the endpoints). Let a_i be a median vertex of $I \cap V^i$ (median based on the ordering), where I is a stage i interval. Define U_{i+1} as the smallest subset of V^i containing $\{a_i\}$ and also $N_H(N_H(U_{i+1}) - V^i) \cap V^i$. Define V_{i+1} as $N_H(U_{i+1}) - V^i$. We now argue that this is indeed a decomposition. Consider a stage i interval I, with s_{j_0}, s_{j_1} as the defining end points, and a_i as the median of $I \cap V^i$.

Lemma 3. *For every $i \geq 1$, there is a stage i interval I with $U_{i+1} \subseteq I$ and $|U_{i+1}| \leq |I \cap V^i| \leq 2\triangle$.*

The properties ① and ③ of the decomposition are guaranteed by the construction. Lemma 3 implies that the width of the decomposition is most $2\triangle^2$. Property ② holds because there are no triangles in H.

4.3 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. *Let H be a planar graph of minimum degree* 2 *and girth at least* 16*, then H always contains a* 3*-thread.*

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

$$X_i = \begin{cases} \{a_i\}, \text{where } a_i \text{ is a degree 0 or 1 vertex in } H[V_H - \bigcup_{j<i} X_j], \textit{or} \\ \{a_i, b_i, c_i\}, \text{where } a_i, b_i, c_i \text{ form a 3-thread in } H[V_H - \bigcup_{j<i} X_j]. \end{cases}$$

Remember that for a subset of vertices S of H, $H[S]$ denotes the subgraph of H induced by S. By Lemma 4, every planar graph with girth at least 16 has a 3-thread partition. As earlier, we say, a vertex is selected if we add it to some V_k. Set $i = 1$. Using the 3-thread partition (which can be constructed using Lemma 4), a decomposition of a planar graph of girth at least 16 can be constructed by repeating the following procedure until all vertices are selected.

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 $V_i = N_H(U_i) - V^{i-1}$.
iii. Increment i.

Lemma 5. *Let H be a planar graph of girth at least 16. Then the above procedure finds a decomposition of H of width at most* $2\triangle$*.*

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 fails to satisfy this condition. For simplicity, we will only consider $\sqrt{n} \times \sqrt{n}$ grid graphs, but our proof extend to other cases as well.

Let $H = (V_H, E_H)$ be a $\sqrt{n} \times \sqrt{n}$ grid graph with $V_H = \{(i,j) : 0 \leq i, j \leq \sqrt{n}-1\}$ 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 any decomposition of H has a width of at least $\sqrt{n}$. Let $V_1, \ldots, V_\ell$ be any decomposition of H. Consider any 2×2 square of H defined by vertices a, b, c, d. The two neighbors a, b of the vertex c with the smallest label l always have the same label $l' > l$. The fourth vertex d has any label l'' with $l'' \geq l$ 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 have the same label thus are chosen together.

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 $0 \leq i, j \leq \sqrt{n} - 2$ 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 top and bottom or left and right, which in turn implies the desired result.

Theorem 5. *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. Therefore, there exists no decomposition of a $\sqrt{n} \times \sqrt{n}$-grid graph H of width $\sqrt{n} - 1$.*

Acknowledgments

We thank Andrzej Ruciński for pointing us to [35] 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 discussions.

References

1. Dong, H., Wu, Y., Ding, X.: An ARG representation for chinese characters and a radical extraction based on the representation. In: International Conference on Pattern Recognition, pp. 920–922 (1988)
2. 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., Taylor, R.: Similarity searching in databases of three-dimensional molecules and macromolecules. Journal of Chemical Information and Computer Sciences 32, 617–630 (1992)
3. Stahs, T., Wahl, F.M.: Recognition of polyhedral objects under perspective views. Computers and Artificial Intelligence 11, 155–172 (1992)
4. Levinson, R.: Pattern associativity and the retrieval of semantic networks. Computers & Mathematics with Applications 23(6–9), 573–600 (1992)
5. Alon, N., Yuster, R., Zwick, U.: Color-coding. Journal of the ACM 42(4), 844–856 (1995)
6. Valiant, L.G.: The complexity of computing the permanent. Theoretical Computer Science 8, 189–201 (1979)
7. Toda, S.: On the computational power of PP and $\oplus P$. In: FOCS, pp. 514–519. IEEE, Los Alamitos (1989)
8. Flum, J., Grohe, M.: The parameterized complexity of counting problems. SIAM Journal of Computing 33(4), 892–922 (2004)
9. Arvind, V., Raman, V.: Approximation algorithms for some parameterized counting problems. In: Bose, P., Morin, P. (eds.) ISAAC 2002. LNCS, vol. 2518, pp. 453–464. Springer, Heidelberg (2002)
10. Kasteleyn, P.W.: Graph theory and crystal physics. In: Harary, F. (ed.). Academic Press, London (1967)
11. Jerrum, M., Sinclair, A., Vigoda, E.: A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. Journal of ACM 51(4), 671–697 (2004)
12. Jerrum, M., Sinclair, A.: Polynomial-time approximation algorithms for the Ising model. SIAM Journal on Computing 22(5), 1087–1116 (1993)
13. Bezáková, I., Bhatnagar, N., Vigoda, E.: Sampling binary contingency tables with a greedy start. In: SODA, pp. 414–423. SIAM, Philadelphia (2006)
14. Alon, N., Frieze, A.M., Welsh, D.: Polynomial time randomized approximation schemes for Tutte-Gröthendieck invariants: The dense case. Random Structures & Algorithms 6(4), 459–478 (1995)
15. Dyer, M., Frieze, A.M., Jerrum, M.: Approximately counting Hamilton paths and cycles in dense graphs. SIAM Journal on Computing 27(5), 1262–1272 (1998)

16. Jerrum, M.: Counting, sampling and integrating: algorithms and complexity. Birkhäuser, Basel (2003)
17. Frieze, A.M., McDiarmid, C.: Algorithmic theory of random graphs. Random Structures & Algorithms 10(1-2), 5–42 (1997)
18. Erdős, P., Rényi, A.: On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci. 5, 17–61 (1960)
19. Jerrum, M., Sinclair, A.: Approximating the permanent. SIAM Journal on Computing 18(6), 1149–1178 (1989)
20. Frieze, A.M., Jerrum, M.: An analysis of a Monte Carlo algorithm for estimating the permanent. Combinatorica 15(1), 67–83 (1995)
21. Rasmussen, L.E.: Approximating the permanent: A simple approach. Random Structures & Algorithms 5(2), 349–362 (1994)
22. Chien, S.: A determinant-based algorithm for counting perfect matchings in a general graph. In: SODA, pp. 728–735. SIAM, Philadelphia (2004)
23. Fürer, M., Kasiviswanathan, S.P.: Approximately counting perfect matchings in general graphs. In: ALENEX/ANALCO, pp. 263–272. SIAM, Philadelphia (2005)
24. Frieze, A.M., Suen, S.: Counting the number of Hamilton cycles in random digraphs. Random Structures & Algorithms 3(3), 235–242 (1992)
25. Frieze, A.M., Jerrum, M., Molloy, M.K., Robinson, R., Wormald, N.C.: Generating and counting Hamilton cycles in random regular graphs. Journal of Algorithms 21(1), 176–198 (1996)
26. Rasmussen, L.E.: Approximately counting cliques. Random Structures & Algorithms 11(4), 395–411 (1997)
27. Hammersley, J.M.: Existence theorems and Monte Carlo methods for the monomer-dimer problem. Research Papers in Statistics, 125–146 (1966)
28. Knuth, D.E.: Estimating the efficiency of backtrack programs. Mathematics of Computation 29(129), 121–136 (1975)
29. Fürer, M., Kasiviswanathan, S.P.: An almost linear time approximation algorithm for the permanent of a random (0-1) matrix. In: Lodaya, K., Mahajan, M. (eds.) FSTTCS 2004. LNCS, vol. 3328, pp. 263–274. Springer, Heidelberg (2004)
30. Sankowski, P.: Alternative algorithms for counting all matchings in graphs. In: Alt, H., Habib, M. (eds.) STACS 2003. LNCS, vol. 2607, pp. 427–438. Springer, Heidelberg (2003)
31. Jerrum, M., Valiant, L., Vazirani, V.: Random generation of combinatorial structures from a uniform distribution. Theoretical Computer Science 43, 169–188 (1986)
32. Luks, E.M.: Isomorphism of graphs of bounded valence can be tested in polynomial time. Journal of Computer and System Sciences 25, 42–65 (1982)
33. Diestel, R.: Graph theory, 2nd edn. Springer, Heidelberg (2000)
34. Fürer, M., Kasiviswanathan, S.P.: Approximately counting embeddings into random graphs. CoRR, arXiv:0806.2287 [cs.DS] (2008)
35. Riordan, O.: Spanning subgraphs of random graphs. Combinatorics, Probability & Computing 9(2), 125–148 (2000)
36. Janson, S., Łuczak, T., Ruciński, A.: Random graphs. Wiley-Interscience, Chichester (2000)