A randomized embedding algorithm for trees

Benny Sudakov* Jan Vondrák †

Abstract

In this paper, we propose a simple and natural randomized algorithm to embed a tree T in a given graph G. The algorithm can be viewed as a "self-avoiding tree-indexed random walk". The order of the tree T can be as large as a constant fraction of the order of the graph G, and the maximum degree of T can be close to the minimum degree of G. We show that our algorithm works in a variety of interesting settings. For example, we prove that any graph of minimum degree d without 4-cycles contains every tree of order ϵd^2 and maximum degree at most $d - 2\epsilon d - 2$. As there exist d-regular graphs without 4-cycles and with $O(d^2)$ vertices, this result is optimal up to constant factors. We prove similar nearly tight results for graphs of given girth and graphs with no complete bipartite subgraph $K_{s,t}$.

1 Introduction

We consider the problem of embedding a tree T in a given graph G. Formally, we look for an injective map $f : V(T) \to V(G)$ which preserves the edges. We do not require that non-edges are mapped to non-edges, i.e. the copy of T in G need not be induced. Our goal is to design a simple algorithm which finds such an embedding, assuming certain conditions on the parameters of T and the graph G.

1.1 Brief history

The problem of embedding paths and trees in graphs has long been one of the fundamental questions in combinatorics. This problem has been extensively studied in extremal combinatorics, in the theory of random graphs, in connection with properties of expanders and with applications to Computer Science. The goal always has been to find a suitable property of a graph G which guarantees that it contains all possible trees with given parameters. We describe next several examples which we think are representative and give a good overview of previous research in this area.

Extremal questions. The basic extremal question about trees is to determine the number of edges that a graph needs to have in order to contain all trees with a given number of edges. It is an old folklore result that a graph G of minimal degree d contains every tree T with d edges. This can be

*Department of Mathematics, UCLA, Los Angeles, CA 90095. Email: bsudakov@math.ucla.edu. Research supported in part by NSF CAREER award DMS-0812005 and by an USA-Israeli BSF grant.

†IBM Almaden Research Center, San Jose, CA 95120. Email: jvondrak@us.ibm.com. This work was done while the author was at Princeton University.

achieved simply by embedding vertices of T greedily one by one. Since at most d vertices of G are occupied at any point, there is always enough room to embed another vertex of the tree.

An old conjecture of Erdős and Sós says that *average degree* d is already sufficient to guarantee the same property. More precisely, any graph with more than $(d-1)n/2$ edges contains all trees with d edges. A clique of order d is an obvious tight example for this conjecture. The conjecture has been proved in several special cases, e.g. Brandt and Dobson [8] establish it for graphs of girth at least 5 (the *girth* is the length of the shortest cycle in a graph). In fact, they prove a stronger statement, that any such graph of minimum degree $d/2$ and maximum degree Δ contains all trees with d edges and maximum degree at most Δ. More generally, improving an earlier result of Haxell and Łuczak [15], Jiang proved that any graph of girth $2k+1$ and minimum degree d/k contains all trees with d edges and maximum degree at most d/k [12]. For general graphs, it has been announced by Ajtai, Komlós, Simonovits and Szemerédi [1] that they proved Erdős-Sós conjecture for all sufficiently large trees.

A related statement, known as Loebl's $(\frac{n}{2}-\frac{n}{2}-\frac{n}{2})$ conjecture [9], is that any graph on n vertices, with at least $n/2$ vertices of degree at least $n/2$, contains all trees with at most $n/2$ edges. Progress on this conjecture has been recently made by Yi Zhao [29]. Note that in the results discussed so far, the number of vertices in the tree is of the same order as the degrees in the graph G. Without assuming any additional properties of G, this seems to be a natural barrier.

Expanding graphs. Embedding trees of order much larger than the average degree of the graph is possible in graphs satisfying certain expansion properties. The first such result was established by Pósa using his celebrated rotation/extension technique. Given a subset of vertices X of a graph G let $N(X)$ denote the set of all neighbors of vertices of X in G. Pósa [25] proved that if $|N(X) \setminus X| \geq 2|X|-1$ for every subset X of G with at most t vertices, then G contains a path of length $3t-2$. This result was extended to trees by Friedman and Pippenger [11]. They proved that if $|N(X)| \geq (d+1)|X|$ for all subsets of size at most $2t-2$, then G contains every tree of order t and maximum degree at most d. The power of this result is that while T can have degrees close to the minimum degree of G, it can be of order much larger than d, depending on the expansion guarantee. An even more powerful application of expansion properties was presented by Haxell and Kohayakawa [14]. It follows from their work that any graph where $|N(X)| \geq 2d|X|$ for all $|X| \leq 2t/d$ contains all trees T of order at most t and degrees bounded by d. Therefore, roughly speaking, it is sufficient to have expansion proportional to d for sets of size $O(|T|/d)$. Using expansion properties, Haxell proved that any graph G of average degree at least $t = 18r$, avoiding a bipartite subgraph $K_{2,r}$, contains all trees of order t [16].

The result of Friedman and Pippenger [11] has other interesting applications. For example, it can be used to show that for a fixed $\delta > 0$, d and every n, there is a graph G with $O(n)$ edges that, even after deleting all but $\delta|E(G)|$ edges, continues to contain every tree with n vertices and maximum degree at most d. This has immediate corollaries in Ramsey Theory. The technique of Friedman and Pippenger also has an application for infinite graphs. For an infinite graph G, its *Cheeger constant* is $h(G) = \inf_X \frac{|N(X)\setminus X|}{|X|}$, where X is a nonempty finite subset of vertices of G. Using the ideas of [11], one can show that any infinite graph G with Cheeger constant $d \geq 3$ contains an infinite tree T with Cheeger constant $d-2$ [5]. Benjamini and Schramm prove a stronger result that any infinite graph with $h(G) > 0$ contains an infinite tree with positive Cheeger constant [5]. They use the notion of *tree-indexed random walks* to find such a tree. We will allude to this notion again later.

Random graphs. The random graph $G_{n,p}$ is a probability space whose points are graphs on a fixed set of n vertices, where each pair of vertices forms an edge, randomly and independently, with probability p. For random graphs, Erdős conjectured that with high probability, $G_{n,d/n}$ for a fixed d contains a very long path, i.e., a path of length $(1-\alpha(d))n$ such that $\lim_{d\to\infty} \alpha(d) = 0$. This conjecture was proved by Ajtai, Komlós and Szemerédi [2] and, in a slightly weaker form, by Fernandez de la Vega [27]. Embedding trees, however, is considerably harder. Fernandez de la Vega [28] showed that there are (large) constants a_1, a_2 such that $G_{n,d/n}$ contains any *fixed* tree T with n/a_1 vertices and maximum degree $\Delta \le d/a_2$ w.h.p (i.e., with probability tending to 1 when $n \to \infty$). Note that this is much weaker than containing all trees simultaneously, because a random graph can contain every fixed tree w.h.p, and still miss at least one tree w.h.p. Until recently, there was no result known on embedding all trees simultaneously. Alon, Krivelevich and Sudakov proved in [2] that for any $\epsilon > 0$, $G_{n,d/n}$ contains all trees with $(1-\epsilon)n$ vertices and maximum degree Δ such that

$$d \ge \frac{10^6}{\epsilon} \Delta^3 \log \Delta \log^2(2/\epsilon).$$

(All logarithms here and in the rest of this paper have natural base.) This result is nearly tight in terms of the number of vertices in T, and holds for all trees simultaneously. However, it is achieved at the price of requiring that degrees in G are much larger than degrees in the tree.

The results of Haxell and Kohayakawa [14], together with the known expansion properties of random graphs, can be used to show that the random graph $G_{n,p}$ with $p = \Omega(\frac{\log n}{n})$ contains with high probability all trees of order $O(n)$ and maximum degree $O(pn)$.

1.2 Our results

We prove several results about our tree embedding algorithm, in particular for graphs G with no short cycles and graphs without a given complete bipartite subgraph $K_{s,t}$. We embed trees with parameters very close to trivial upper bounds that cannot be exceeded: maximum degree is close to the minimum degree of G, and the number of vertices is a constant fraction of the order of G (or more precisely, the minimum possible order of G under given conditions). The *order* of a graph is the number of its vertices and the *size* is the number of its edges. For trees, the two quantities differ only by 1. A summary of our main results follows. Here we assume that d and n are sufficiently large.

1. For any constant $k \ge 2$, $\epsilon \le \frac{1}{2k}$ and any graph G of girth at least $2k+1$ and minimum degree d, G contains every tree T of order $|T| \le \frac{1}{4}\epsilon d^k$ and maximum degree $\Delta \le (1-2\epsilon)d - 2$.

2. For any G of minimum degree d, not containing $K_{s,t}$ (the complete bipartite graph with parts of size $s \ge t \ge 2$), G contains every tree T of order $|T| \le \frac{1}{64 s^{1/(t-1)}} d^{1+\frac{1}{t-1}}$ and maximum degree $\Delta \le \frac{1}{256} d$.

It is easy to see that any graph of girth $2k+1$ and minimum degree d has at least $\Omega(d^k)$ vertices. It is a major open question to determine the smallest possible order of such graph. For values of $k = 2, 3, 5$ there are known constructions obtained by Erdős and Rényi [10] and Benson [6] of graphs of girth $2k+1$, minimum degree d and order $O(d^k)$. It is also widely believed that such constructions should be possible for all fixed k. This implies that our first statement is tight up to constant factors for $k = 2, 3, 5$ and probably for all remaining k. Similarly, it is conjectured that for $s \geq t$ there are $K_{s,t}$-free graphs with minimum degree d which have $O(d^{1+\frac{1}{t-1}})$ vertices. For $s > (t-1)!$, such a construction was obtained by Alon, Rónyai and Szabo [3] (modifying the construction in [17]). Hence, the order of the trees we are embedding in our second result is tight up to constant factors as well.

1.3 Discussion

Local expansion. Using well-known results from extremal graph theory, one can show that if a graph G contains no subgraphs isomorphic to a fixed bipartite graph H (e.g., C_{2k} or $K_{s,t}$), then it has certain expansion properties. More precisely, all small subsets of G have a large boundary. For example, if G is a C_4-free graph with minimum degree d then all subsets of G of size at most d expand by a factor of $\Theta(d)$. Otherwise we would get a 4-cycle by counting the number of edges between S and its boundary $N(S) \setminus S$. This simple observation appears to be a powerful tool in attacking various extremal problems and was used in [26] and [21] to resolve several conjectures about cycle lengths and clique-minors in H-free graphs.

Therefore, it is natural to ask whether the expansion of H-free graphs combined with the result of Friedman and Pippenger can be used to embed large trees. Recall that to embed a tree with t vertices and maximum degree d, Friedman and Pippenger require that sets of size up to $2t-2$ expand at least $d+1$ times. Plugging this into the observation we made on the expansion of C_4-free graphs only gives embedding of trees of order $O(d)$ in such graphs. However, to apply the more powerful result of Haxell and Kohayakawa [14], it is sufficient to have expansion for sets of size $O(t/d)$. Therefore, using this result it can be shown that C_4-free graphs contain all trees with $O(d^2)$ vertices, which is the same order of magnitude that we prove.

One can also use the result of Haxell and Kohayakawa to embed large trees in $K_{s,t}$-free graphs. Indeed, it is not hard to show (see, e.g., [21]) that if G is a $K_{s,t}$-free graph ($s \geq t$) with minimum degree d, then all subsets of G of size at most $d^{1/(t-1)}$ expand by a factor of $\Theta(d/s)$. Therefore, $K_{s,t}$-free graphs with minimum degree d contain all trees of order $\Omega\big(d^{1+1/(t-1)}\big)$ and maximum degree $O(d/s)$. In particular, for large s the maximum degree of the tree in this result is only a tiny fraction of the degree of G. On the other hand, we embed trees whose maximum degree is proportional to d with a factor that is independent from s.

For $k \geq 3$ and C_{2k}-free graphs, we also achieve better results than what follows from expansion properties. Using the best known bounds on the number of edges in a C_{2k}-free graph from [24], it was shown in [21] that any set of vertices of size $O\big(d^{\lfloor k/2 \rfloor}\big)$ in a C_{2k}-free graph with minimum degree d expands by a factor of $\Omega(d)$. Using the result of Haxell and Kohayakawa [14], this implies that C_{2k}-free graphs with minimum degree d contain all trees of order $O(d^{\lfloor k/2+1 \rfloor})$ with degrees bounded by $O(d)$. However, we can embed trees of order up to $O(d^k)$, which is tight modulo the well known conjecture in extremal graph theory.

Extremal results. Our work sheds some light on why the Erdős-Sós conjecture, which we already discussed in the beginning of the introduction, becomes easier for graphs with no short cycles or no complete bipartite subgraphs $K_{2,r}$. These scenarios were considered, e.g., in [8, 15, 12, 16]. In particular, assuming that graph G has girth $2k+1, k \geq 2$, and minimum degree d, Jiang [12] showed how to embed in G all trees with kd vertices and degrees bounded by d. Although this is best possible, our result implies that this statement can be tight only for relatively few very special trees, i.e., those that contain several large stars of degree extremely close to d. Indeed, if we relax the degree assumption and consider trees with maximum degree at most $(1-\epsilon)d$, then it is possible to embed trees of order $O(d^k)$ rather than $O(d)$. Moreover, a careful analysis of our proof shows that it still works when ϵ has order of magnitude $k\frac{\log d}{d}$. Therefore, even if we allow the degree of the tree to be as large as $d - ck \log d$ for some constant c, we are still able to embed all trees of order $\Omega(kd^{k-1} \log d) \gg kd$.

1.4 The algorithm

All our results are proved using variants of the following very simple randomized embedding algorithm. First, arbitrarily choose some vertex r of T to be the *root*. Then for every other vertex $u \in V(T)$ there is a unique path in T from r to u. The neighbor of u on this path is called the *parent* of u and all the remaining neighbors of u are called *children* of u. The algorithm proceeds as follows.

Algorithm 1. *Start by embedding the root r at an arbitrary vertex $f(r) \in V(G)$. As long as T is not completely embedded, take an arbitrary vertex $u \in V(T)$ which is already embedded, but whose children are not. If $f(u)$ has enough neighbors in G unoccupied by other vertices of T, embed the children of u by choosing vertices uniformly at random from the available neighbors of $f(u)$ and continue. Otherwise, fail.*

This algorithm can be seen as a variant of a *tree-indexed random walk*, i.e. a random process corresponding to a tree where each vertex assumes a random state depending only on the state of its parent. The notion of a tree-indexed random walk was first introduced and studied by Benjamini and Peres [4]. It is also used in the above mentioned paper of Benjamini and Schramm [5] to embed trees with a positive Cheeger constant into infinite expanding graphs. In our case, we consider in fact a *self-avoiding* tree-indexed random walk, where each state is chosen randomly, conditioned on being distinct from previously chosen states. The corresponding concept for a random walk is a well studied subject in probability (see, e.g., [22]). Loosely speaking, we prove that our self-avoiding tree-indexed random walk behaves sufficiently randomly, in the sense that it does not intersect the neighborhood of any vertex more often than expected. To analyze the number of times the random process intersects a given neighborhood, we use large deviation inequalities for supermartingales.

1.5 A supermartingale tail estimate

In all our proofs, we use the following tail estimate.

Proposition 1.1 *Let $X_1, X_2, \ldots, X_n$ be random variables in $[0,1]$ such that for each k,*

$$\mathbb{E}[X_k \mid X_1, X_2, \ldots, X_{k-1}] \leq a_k.$$

Let $\mu = \sum_{k=1}^n a_k$. Then for any $0 < \delta \leq 1$,

$$\mathbb{P}[\sum_{k=1}^n X_k > (1+\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{3}}.$$

This can be derived easily from the proof of Theorem 3.12(b) in [23]. We re-state this theorem here: *Let $Y_1, Y_2, \dots, Y_n$ be a martingale difference sequence with $-a_k \le Y_k \le 1 - a_k$ for each k, for suitable constants a_k, and let $a = \frac{1}{n}\sum a_k$. Then for any $\delta > 0$,*

$$\mathbb{P}[\sum_{k=1}^{n} Y_k \ge \delta a n] \le e^{-\frac{\delta^2 a n}{2(1+\delta/3)}}.$$

A martingale difference sequence satisfies $\mathbb{E}[Y_k \mid Y_1, Y_2, \dots, Y_{k-1}] = 0$. However, it can be seen easily from the proof in [23] that for this one-sided tail estimate, it is sufficient to assume $\mathbb{E}[Y_k \mid Y_1, Y_2, \dots, Y_{k-1}] \le 0$. (Such a random process is known as a *supermartingale*.) To show Proposition 1.1, set $Y_k = X_k - a_k$ and $\mu = an = \sum_{k=1}^{n} a_k$. The conditional expectations of X_k are bounded by a_k, hence the conditional expectations of Y_k are non-positive as required. Since $\delta \le 1$, we also replace $2(1 + \delta/3)$ by 3, and Proposition 1.1 follows.

Note also that we can always replace μ by a larger value (e.g., by adding auxiliary random variables that are constants with probability 1), and the conclusion still holds. Hence, in Proposition 1.1 it is enough to assume $\sum_{k=1}^{n} a_k \le \mu$.

2 Embedding trees in C_4-free graphs

The purpose of this section is to illustrate on a simple example the main ideas and techniques that we will use in our proofs. We start with C_4-free graphs, which is a special case of two classes of graphs we are interested in: graphs without short cycles, and graphs without $K_{s,t}$ (note that $K_{2,2} = C_4$).

Let us recall Algorithm 1. For a given rooted tree T, we start by embedding the root $r \in V(T)$ at an arbitrary vertex $f(r) \in V(G)$. As long as T is not completely embedded, we take an arbitrary $u \in V(T)$ which is already embedded but whose children are not. If $f(u)$ has enough unoccupied neighbors in G, we embed the children of u uniformly at random in the available neighbors of $f(u)$ and continue. Otherwise, we fail.

Theorem 2.1 *Let $\epsilon \le 1/8$, and let G be a C_4-free graph of minimum degree at least d. For any tree T of order $|T| \le \epsilon d^2$ and maximum degree $\Delta \le d - 2\epsilon d - 2$, Algorithm 1 finds an embedding of T in G with high probability (i.e., with probability tending to 1 when $d \to \infty$).*

Example. Before we plunge into the proof, let us consider the statement of this theorem in a particular case, where G is the incidence graph of a finite projective plane. Let $q = d - 1$ be a prime power and consider the 3-dimensional vector space over the finite field $\mathbb{F}_q$. Let V_1 be all 2-dimensional linear subspaces of $\mathbb{F}_q^3$ (lines in a projective plane), V_2 all 1-dimensional linear subspaces (points in a projective plane) and let two vertices from V_1 and V_2 be adjacent if their corresponding subspaces contain one another. This G has $n = 2(q^2 + q + 1) = 2(d^2 - d + 1)$ vertices, and it is bipartite and d-regular. Also, it is easy to see from the definition that G does not contain C_4. Clearly, we cannot embed in G trees of order larger than $O(d^2)$ or maximum degree larger than d. In this respect, our theorem is tight up to constant factors.

It is also worth mentioning that in the analysis of our simple algorithm, the trade-off between the number of vertices in T and the maximum degree Δ is close to being tight. Indeed, we show that for $\Delta = (1-\epsilon)d$, our algorithm cannot embed trees of order much larger than ϵd^2. Suppose we are embedding a tree T of depth 3, where the degrees of the root and its children (level 1) are $\sqrt{d}$. On level 2, the degrees are ϵd except one special vertex z of degree $(1-\epsilon)d$. On level 3, there are only leaves. The number of vertices in this tree is $\epsilon d^2 + \Theta(d)$.

We can assume that the root is embedded at a vertex corresponding to a point a. The level-1 vertices are embedded into a set L_1 of $\sqrt{d}$ random lines through a. The level-2 vertices are embedded into a set P_2 of d random points on these lines. Every point in the projective plane (except a) has the same probability of appearing in P_2, hence this probability is $d/(d^2-d) = 1/(d-1)$. The level-3 vertices are embedded into random lines L_3 through points in P_2, each line through a point in P_2 with probability ϵ. Now every line has probability roughly ϵ of being in L_3, because on average, one of its points appears in P_2. Consider the point where we embed the special vertex z and assume that this is the last vertex we process in the algorithm. Each of the d lines through this point has probability roughly ϵ of being occupied by a level-3 vertex, so on average, only $(1-\epsilon)d$ lines are available to host the children of z. Therefore, our algorithm cannot succeed in embedding more than $(1-\epsilon)d$ children of z.

Proof of Theorem 2.1. Let us fix an ordering in which the algorithm processes the vertices of T: $V(T) = \{1, 2, \ldots, |V(T)|\}$. Here, 1 denotes the root and the ordering is consistent with the structure of the tree in the sense that every vertex appears only after its parent. In step 0, the algorithm embeds the root. In step t, the children of t are embedded randomly in the yet unoccupied neighbors of $f(t) \in V(G)$. If t is a leaf in T, the algorithm is idle in step t.

Our goal is to argue that for large d, with high probability, the algorithm never fails. The only way the algorithm can fail is that for a vertex $t \in V(T)$ embedded at $v = f(t) \in V(G)$, we are not able to place its children because too many neighbors of v in G have been occupied by other vertices of T. This is the crucial "bad event" we have to analyze:

Let $\mathcal{B}_v$ denote the event that at some point, more than $2\epsilon d + 2$ neighbors of v are occupied by vertices of T other than the children of $f^{-1}(v)$.

If we can show that with high probability, $\mathcal{B}_v$ does not occur for any $v \in V(G)$, then the algorithm clearly succeeds. To do this, we will modify our algorithm slightly and force it to stop immediately at the moment when the first bad event occurs. Thus, in analyzing $\mathcal{B}_v$, we can assume that for any $w \neq v$ the event $\mathcal{B}_w$ has not happened yet.

Our strategy is to prove that the probability of $\mathcal{B}_v$ for any given vertex v, even conditioned on our embedding getting "dangerously close" to v, is exponentially small in d. Then, we argue that the number of vertices which can ever get dangerously close to our embedding (i.e., the number of bad events we have to worry about) is only polynomial in d. Therefore, we conclude that with high probability, no bad event occurs.

Lemma 2.2 *Let $\epsilon \leq \frac{1}{8}$ and $d \geq 24$. For a vertex $v \in V(G)$, condition on any history $\mathcal{H}$ of running the algorithm up to a certain point such that at most 2 vertices of T have been embedded in $N(v)$. Then*

$$\mathbb{P}[\mathcal{B}_v \mid \mathcal{H}] \leq e^{-\epsilon d/18}.$$

Proof. For $t = 1, 2, \ldots, |V(T)|$, let X_t be the indicator variable of the event that $f(t) \neq v$ but some child of t gets embedded in $N(v)$. Here we use the property that G is C_4-free. Note that if

$f(t) = w \neq v$, w can have at most one neighbor in $N(v)$, or else we have a 4-cycle. Therefore, t can have at most one child embedded in $N(v)$ and X_t represents the number of vertices in $N(v)$, which are occupied by the children of t.

We condition on a history $\mathcal{H}$ of running the algorithm up to step h, such that at most 2 vertices of $N(v)$ have been occupied so far. The bad event $\mathcal{B}_v$ can occur only if $X = \sum_{t=h+1}^{|T|} X_t > 2\epsilon d$. Therefore, our goal is to prove that this happens only with very small probability.

Each vertex chooses the embedding of its children randomly, out of at least $d - 2\epsilon d - 2$ still available choices (here we assume that no bad event B_w occurred before $\mathcal{B}_v$ for any $w \neq v$, or else the algorithm has failed already). Thus we get

$$\mathbb{E}[X_t] \leq \frac{d_T(t)}{d - 2\epsilon d - 2} \leq \frac{d_T(t)}{2d/3}$$

where $d_T(t)$ is the number of children of the vertex t in T. We also used $\epsilon \leq 1/8$ and $d \geq 24$. This holds even conditioned on any previous history of the algorithm, since the decisions for each vertex are made independently. We are interested in the probability that $X = \sum_{t=h+1}^{|T|} X_t$ exceeds $2\epsilon d$. Using the fact that $\sum_{t \in T} d_T(t) = |T| - 1 \leq \epsilon d^2$, we can bound the expectation of X by

$$\mu = \mathbb{E}[X] = \sum_{t=h+1}^{|T|} \mathbb{E}[X_t] \leq \sum_{t \in T} \frac{d_T(t)}{2d/3} \leq \frac{3}{2}\epsilon d.$$

We use the supermartingale tail estimate (Proposition 1.1) with $\delta = \frac{1}{3}$ and $\mu = \frac{3}{2}\epsilon d$:

$$\mathbb{P}[X > 2\epsilon d] \leq e^{-\delta^2 \mu/3} = e^{-\mu/27} = e^{-\epsilon d/18}.$$

Therefore, the bad event $\mathcal{B}_v$ happens with probability at most $e^{-\epsilon d/18}$. □

Our final goal is to argue that with high probability, no bad event $\mathcal{B}_v$ occurs for any vertex $v \in V(G)$. Since the number of vertices could be potentially unbounded by any function of d, we cannot apply a straightforward union bound over all vertices in the graph. However, we observe that the number of vertices for which $\mathcal{B}_v$ can potentially occur is not very large.

Define $\mathcal{D}_v$ to be the event that at some point in the algorithm, two vertices in $N(v)$ are occupied by vertices of T. This is the event that the embedding of T gets "dangerously close" to v. Observe that if $\mathcal{D}_v$ is "witnessed" by the pair of vertices of T which are placed in $N(v)$, each pair of vertices of T can witness at most one event $\mathcal{D}_v$ (otherwise the same pair is in the neighborhood of two vertices which implies a C_4). Since T has at most ϵd^2 vertices, the event $\mathcal{D}_v$ can occur for at most $\epsilon^2 d^4$ vertices in any given run of the algorithm.

Clearly, the events $\mathcal{B}_v$ satisfy $\mathcal{B}_v \subseteq \mathcal{D}_v$. Let us analyze the probability of $\mathcal{B}_v$, conditioned on $\mathcal{D}_v$. The event $\mathcal{D}_v$ can be written as a union of all histories $\mathcal{H}$ of running the algorithm up to the point where two vertices of T get embedded in $N(v)$. By Lemma 2.2,

$$\mathbb{P}[\mathcal{B}_v \mid \mathcal{H}] < e^{-\epsilon d/18}$$

for any such history $\mathcal{H}$. By taking the union of all these histories, we get

$$\mathbb{P}[\mathcal{B}_v \mid \mathcal{D}_v] < e^{-\epsilon d/18}.$$

Now we can estimate the probability that $\mathcal{B}_v$ ever occurs for any vertex v:

$$\mathbb{P}[\exists v \in V; \mathcal{B}_v \text{ occurs}] \leq \sum_{v \in V} \mathbb{P}[\mathcal{B}_v] = \sum_{v \in V} \mathbb{P}[\mathcal{B}_v \mid \mathcal{D}_v]\mathbb{P}[\mathcal{D}_v] \leq e^{-\epsilon d/18} \sum_{v \in V} \mathbb{P}[\mathcal{D}_v].$$

Since $\mathcal{D}_v$ can occur for at most $\epsilon^2 d^4$ vertices in any given run of the algorithm, we have $\sum_{v \in V} \mathbb{P}[\mathcal{D}_v] \leq \epsilon^2 d^4$. Thus

$$\mathbb{P}[\exists v \in V; \mathcal{B}_v \text{ occurs}] \leq \epsilon^2 d^4 e^{-\epsilon d/18} \to 0,$$

when $d \to \infty$. Hence the algorithm succeeds with high probability. □

3 Embedding trees in $K_{s,t}$-free graphs

Next, we consider the case of graphs which contain no complete bipartite subgraph $K_{s,t}$ with parts of size s and t. We assume that $s \geq t$. It is known that the extremal size of such graphs depends essentially only on the value of the smaller parameter t. Indeed, by the result of Kövari, Sós and Turán [18], the number of vertices in a $K_{s,t}$-free graph with minimum degree d is at least $c\, d^{t/(t-1)}$, where only the constant c depends on s. For relatively high values of s ($s > (t-1)!$) there are known constructions (see, e.g., [17, 3]) of $K_{s,t}$-free graphs achieving this bound. Moreover, it is conjectured that $\Theta(d^{t/(t-1)})$ is the correct bound for all $s \geq t$. This implies that one cannot embed trees larger than $O(d^{t/(t-1)})$ in a $K_{s,t}$-free graph with minimum degree d. Also, it is obvious that the maximum degree in the tree should be $O(d)$. In this section we show how to embed trees with parameters very close to these natural limits. It is easier to analyze our algorithms in the case when the maximum degrees in the tree are in fact bounded by $O(d/t)$. First, we obtain this weaker result, and then present a more involved analysis which shows that our algorithm also works for trees with maximum degree at most $\frac{1}{256}d$. Our algorithm here is a slight modification of Algorithm 1.

Algorithm 2. *For each vertex $v \in V(G)$, fix a set of d neighbors $N_+(v) \subseteq N(v)$. Start by embedding the root of the tree $r \in T$ at an arbitrary vertex $f(r) \in V(G)$. As long as T is not completely embedded, take an arbitrary vertex $u \in V(T)$ which is already embedded but whose children are not. If $f(u)$ has enough neighbors in $N_+(f(u))$ unoccupied by other vertices of T, embed the children of u one by one, by choosing vertices uniformly at random from the available vertices in $N_+(f(u))$, and continue. Otherwise, fail.*

The only difference from the original algorithm is that when embedding the children of a vertex, we choose from a predetermined set of d neighbors rather than all possible neighbors. Since the maximum degree of G can be very large, this modification is useful in the analysis of our algorithm. It allows us to bound the number of dangerous events. However, we believe that the original algorithm works as well and only our proof requires this modification.

Theorem 3.1 *Let G be a $K_{s,t}$-free graph ($s \geq t$) with minimum degree d. For any tree T of order $|T| \leq \frac{1}{64}s^{-1/(t-1)}d^{t/(t-1)}$ and maximum degree $\Delta \leq \frac{1}{64t}d$, Algorithm 2 finds an embedding of T in G with high probability.*

Proof. We follow the strategy of defining bad events for each vertex $v \in V(G)$ and bounding the probability that any such event occurs.

Let $\mathcal{B}_v$ denote the event that at some stage of the algorithm, more than $\frac{1}{2}d + 2t$ vertices in $N_+(v)$ are occupied by vertices of T other than children of $f^{-1}(v)$.

Note that (as in the previous section) to bound the probability of a bad event, we assume that our algorithm stops immediately at the moment when the first such event occurs. To simplify our analysis, we also assume that the children of every vertex of T are embedded in some particular order, one by one. As long as $\mathcal{B}_v$ does not occur, we have at least $\frac{1}{2}d - 2t$ unoccupied vertices in $N_+(v)$. Since all degrees in the tree are bounded by $\frac{1}{64t}d \le \frac{1}{64}d$, we have enough space for the children of any vertex to be embedded in $N_+(v)$. As we embed the children one by one, the last child still has at least $\frac{1}{2}d - 2t - \frac{1}{64}d \ge \frac{1}{4}d$ choices available (for large enough d).

The new complication here is that another vertex w could share many neighbors with v. Unlike in the case of $K_{2,2}$-free graphs, where any two vertices can share at most 1 neighbor, in $K_{s,t}$-free graphs (for $s \ge t > 2$), we do not have any bound on the number of shared neighbors. Therefore we have to proceed more carefully. For every vertex v in G, we partition all other vertices into two sets depending on how many neighbors they have in $N_+(v)$:

- $L_v = \{w \neq v : |N_+(v) \cap N_+(w)| \le 2s^{\frac{1}{t-1}} d^{\frac{t-2}{t-1}}\}$.
- $M_v = \{w \neq v : |N_+(v) \cap N_+(w)| > 2s^{\frac{1}{t-1}} d^{\frac{t-2}{t-1}}\}$.

The idea is that the vertices in L_v are harmless because the fraction of their children that affect $N_+(v)$ is $O(d^{-1/(t-1)})$. Since the trees we are embedding have $O(d^{1+1/(t-1)})$ vertices, we show that the expected impact of these children on $N_+(v)$ is $O(d)$.

The vertices in M_v have to be treated in a different way, because the fraction of their children in $N_+(v)$ could be very large. However, we prove that the total number of edges between M_v and $N_+(v)$ cannot be too large, otherwise we would get a copy of $K_{s,t}$ in G. Therefore, the impact of the children of M_v on $N_+(v)$ can be also controlled. Again, we "start watching" a bad event for vertex v only at the moment when it becomes dangerous.

Let $\mathcal{D}_v$ denote the event that at least t vertices in $N_+(v)$ are occupied by vertices of the tree T which are not children of $f^{-1}(v)$.

Lemma 3.2 *Let $\mathcal{H}$ be a fixed history of running the algorithm up to a point where at most t vertices in $N_+(v)$ are occupied. Conditioned on $\mathcal{H}$, the probability that the children of vertices embedded in L_v will ever occupy more than $\frac{1}{4}d + t$ vertices in $N_+(v)$ is at most $e^{-d/24}$.*

Proof. We use an argument similar to the proof of Lemma 2.2. Fix an ordering of the vertices of T starting from the root, $i = 1, 2, \ldots, |T|$, as they are processed by the algorithm. Suppose that vertices $1, \ldots, h$ were embedded during the history $\mathcal{H}$. Let X_i be the indicator variable of the event that $i \in T$ is embedded in $N_+(v)$ and the parent of i was embedded in L_v. As long as the algorithm does not fail (i.e., no bad event happened), for each vertex $i \in T$, when it is embedded we have at least $d - \frac{1}{2}d - 2t - \frac{1}{64}d \ge \frac{1}{4}d$ choices for where to place the vertex. This holds even if we condition on any fixed embedding of the vertices $j < i$. Moreover, the embedding decisions for different vertices are independent. Since we assume that the parent of i was embedded in L_v, at most $2s^{1/(t-1)}d^{(t-2)/(t-1)}$

of these choices are in $N_+(v)$. Therefore, conditioned on any previous history $\mathcal{H}$ such that i was not embedded yet

$$\mathbb{P}[X_i = 1 \mid \mathcal{H}] \le \frac{2s^{\frac{1}{t-1}} d^{\frac{t-2}{t-1}}}{\frac{1}{4}d} = 8\left(\frac{s}{d}\right)^{\frac{1}{t-1}}.$$

Summing up over such vertices i in the tree, whose number is at most $|T| \le \frac{1}{64}s^{-1/(t-1)}d^{t/(t-1)}$, we have

$$\mathbb{E}\Big[\sum_{i=h+1}^{|T|} X_i \mid \mathcal{H}\Big] = \sum_{i=h+1}^{|T|} \mathbb{E}[X_i = 1 \mid \mathcal{H}] \le |T| \cdot 8\left(\frac{s}{d}\right)^{\frac{1}{t-1}} \le \frac{1}{8}d.$$

Since, the upper bound on $\mathbb{P}[X_i = 1 \mid \mathcal{H}]$ is still valid even if we also condition on a fixed embedding of all vertices $j < i$, by Proposition 1.1 with $\mu = \frac{1}{8}d$ and $\delta = 1$,

$$\mathbb{P}\Big[\sum_{i=h+1}^{|T|} X_i > \frac{1}{4}d \mid \mathcal{H}\Big] < e^{-d/24}.$$

By definition of $\mathcal{H}$, during the first h steps of the algorithm only at most t vertices in $N_+(v)$ have been occupied. Therefore, the probability that more than $\frac{1}{4}d + t$ vertices are ever occupied is at most $e^{-d/24}$. □

Next, we treat the vertices whose parent is embedded in M_v. Recall that each vertex in M_v has many neighbors in $N_+(v)$. However, the number of edges between M_v and $N_+(v)$ cannot be too large. Observe that there is no $K_{s,t-1}$ in G with s vertices in $N_+(v)$ and $t-1$ vertices in M_v, or else we would obtain a copy of $K_{s,t}$ by adding v to the part of size $t-1$. Also, this shows that for $t = 2$, M_v must be empty. Indeed, by definition any vertex in M_v has at least $2s$ neighbors in $N_+(v)$, which together with vertex v would form $K_{2s,2}$. So in the following, we can assume $s \ge t \ge 3$. The following is a standard estimate in extremal graph theory, whose short proof we include here for the sake of completeness.

Lemma 3.3 *Consider a subgraph H_v containing the edges between M_v and $N_+(v)$, where $|N_+(v)| = d$, every vertex in M_v has at least $2s^{1/(t-1)}d^{(t-2)/(t-1)}$ neighbors in $N_+(v)$, and the graph does not contain $K_{s,t-1}$ (with s vertices in $N_+(v)$ and $t-1$ vertices in M_v). Then H_v has at most $2td$ edges.*

Proof. Let m denote the number of edges in H_v, and assume $m > 2td$. Let N denote the number of copies of $K_{1,t-1}$ (a star with $t-1$ edges) in H_v, with 1 vertex in $N_+(v)$ and $t-1$ vertices in M_v. By convexity, the minimum number of $K_{1,t-1}$ in H_v is attained when all vertices in $N_+(v)$ have the same degree $m/|N_+(v)|$. Therefore

$$N \ge |N_+(v)| \binom{\frac{m}{|N_+(v)|}}{t-1} = d\binom{\frac{m}{d}}{t-1}.$$

Our assumption that $m > 2td$ implies that $\frac{m}{d}, \frac{m}{d} - 1, \ldots, \frac{m}{d} - (t-2) \ge \frac{m}{2d}$ and therefore

$$N \ge d\frac{(\frac{m}{2d})^{t-1}}{(t-1)!} = \frac{m^{t-1}}{(t-1)!2^{t-1}d^{t-2}}.$$

Since all the degrees in M_v are at least $2s^{1/(t-1)}d^{(t-2)/(t-1)}$, we have $m \ge 2s^{1/(t-1)}d^{(t-2)/(t-1)}|M_v|$. Then $m^{t-1} \ge 2^{t-1}sd^{t-2}|M_v|^{t-1}$ and

$$N \ge \frac{m^{t-1}}{(t-1)!2^{t-1}d^{t-2}} \ge \frac{s|M_v|^{t-1}}{(t-1)!} \ge s\binom{|M_v|}{t-1}.$$

Consequently, there must be a $(t-1)$-tuple in M_v which appears in at least s copies of $K_{1,t-1}$. This creates a copy of $K_{s,t-1}$, contradiction. □

Lemma 3.4 *Let $\mathcal{H}$ be a fixed history of running the algorithm up to a point where at most t vertices in $N_+(v)$ are occupied. Then, conditioned on $\mathcal{H}$, the probability the that children of vertices embedded in M_v will ever occupy more than $\frac{1}{4}d+t$ vertices in $N_+(v)$ is at most $t\sqrt{d}e^{-\frac{1}{24t}\sqrt{d}}$.*

Proof. As we mentioned, we can assume $s \geq t \geq 3$, or else M_v is empty. Consider the vertices in M_v, and for every $w \in M_v$, denote the number of edges from w to $N_+(v)$ by d_w. We know that each vertex $w \in M_v$ has $d_w \geq 2s^{1/(t-1)}d^{(t-2)/(t-1)} \geq 2\sqrt{d}$ (using $t \geq 3$). From Lemma 3.3, we know that the total number of these edges is $\sum_{w \in M_v} d_w \leq 2td$. This implies that $|M_v| \leq 2td/(2\sqrt{d}) \leq t\sqrt{d}$.

Suppose we have already seen the history $\mathcal{H}$, and consider $w \in M_v$. Counting from this point onwards, let X_w denote the number of tree vertices embedded in $N_+(v)$, whose parent is embedded at w. We claim that with high probability, $X_w \leq \frac{1}{8t}d_w$. This can be seen as follows. Suppose that $f(x) = w$ for some $x \in V(T)$. The degree of x in T is at most $\frac{1}{64t}d$ and the children of x are embedded one by one. Hence as we already explained, if no bad event $\mathcal{B}_w$ happened so far, each child y has at least $\frac{1}{4}d$ choices available for its embedding. Therefore, even conditioned on the embedding of the previous children, the probability that y is embedded in $N_+(v)$ is at most $p = \min\{1, d_w/(\frac{1}{4}d)\}$. So X_w satisfies the conditions of Proposition 1.1 with $\mu = \frac{1}{64t}d \cdot d_w/(\frac{1}{4}d) = \frac{1}{16t}d_w$. By Proposition 1.1 with $\delta = 1$,

$$\mathbb{P}[X_w > \frac{1}{8t}d_w] \leq e^{-\mu/3} = e^{-\frac{1}{48t}d_w} \leq e^{-\frac{1}{24t}\sqrt{d}},$$

using $d_w \geq 2\sqrt{d}$. By the union bound, the probability that $X_w > \frac{1}{8t}d_w$ for some $w \in M_v$ is at most $|M_v|e^{-\frac{1}{24t}\sqrt{d}} \leq t\sqrt{d}e^{-\frac{1}{24t}\sqrt{d}}$. Otherwise,

$$\sum_{w \in M_v} X_w \leq \frac{1}{8t}\sum_{w \in M_v} d_w \leq \frac{1}{8t} \cdot 2td = \frac{1}{4}d.$$

Together with the t vertices possibly occupied within the history $\mathcal{H}$, this gives at most occupied $\frac{1}{4}d+t$ vertices in $N_+(v)$. □

Having finished all of the necessary preparations, we are now ready to complete the proof of Theorem 3.1. The bad event $\mathcal{B}_v$ can occur only if more than $\frac{1}{4}d+t$ vertices are occupied in $N_+(v)$ by children of vertices in L_v, or more than $\frac{1}{4}d+t$ vertices by children of vertices in M_v. As we proved, each of these events has probability smaller than $t\sqrt{d}e^{-\sqrt{d}/(24t)}$. Therefore, the probability of $\mathcal{B}_v$ is at most $2t\sqrt{d}e^{-\sqrt{d}/(24t)}$. This holds even if we condition on the event $\mathcal{D}_v$ (a disjoint union of histories $\mathcal{H}$) which occurs at the moment when t vertices in $N_+(v)$ are occupied.

Let us estimate the number of events $\mathcal{D}_v$ which can occur. The event $\mathcal{D}_v$ is witnessed by a t-tuple of vertices of the tree T which are embedded in $N_+(v)$. The same t-tuple cannot be a witness for s different events $\mathcal{D}_v$, because then we would have a copy of $K_{s,t}$ in our graph G. Therefore, each t-tuple can witness at most $s-1$ events and the total number of events $\mathcal{D}_v$ is bounded by $(s-1)|T|^t \leq sd^{2t}$. Since $\mathcal{D}_v$ can occur for at most sd^{2t} vertices in any given run of the algorithm, we have $\sum_{v \in V} \mathbb{P}[\mathcal{D}_v] \le sd^{2t}$. Thus

$$\begin{aligned} \mathbb{P}[\exists v \in V; \mathcal{B}_v \text{ occurs}] &\le \sum_{v \in V} \mathbb{P}[\mathcal{B}_v] = \sum_{v \in V} \mathbb{P}[\mathcal{B}_v \mid \mathcal{D}_v]\mathbb{P}[\mathcal{D}_v] \\ &\le 2t\sqrt{d} e^{-\frac{1}{24t}\sqrt{d}} \sum_{v \in V} \mathbb{P}[\mathcal{D}_v] \le 2st\, d^{2t+\frac{1}{2}} e^{-\frac{1}{24t}\sqrt{d}} \end{aligned}$$

which tends to 0 as $d \to \infty$. □

Finally, we show how to prove the same result for trees whose degrees can be a constant fraction of d, independent of t. The following is a strengthened version of Theorem 3.1.

Theorem 3.5 *Let G be a $K_{s,t}$-free graph ($s \ge t$) of minimum degree d. For any tree T of order $|T| \le \frac{1}{64} s^{-1/(t-1)} d^{t/(t-1)}$ and maximum degree $\Delta \le \frac{1}{256} d$, Algorithm 2 finds an embedding of T in G with high probability.*

Proof. The proof is very similar to the proof of Theorem 3.1, with some additional ingredients. We can assume that $t \ge 5$, or else the result follows directly from Theorem 3.1. We focus on the new issues which arise from the fact that the degrees in the tree can exceed $O(d/t)$. For a fixed vertex v, consider again the set M_v defined by

$$M_v = \{w \ne v : |N_+(v) \cap N_+(w)| > 2s^{\frac{1}{t-1}} d^{\frac{t-2}{t-1}}\}.$$

We know from Lemma 3.3 that the number of edges from M_v to $N_+(v)$ is bounded by $2td$. Before, we argued that since the degrees are bounded by $O(d/t)$, the expected contribution from vertices embedded along edges from M_v to $N_+(v)$ cannot be too large. The vertices in T that could cause trouble are those embedded in M_v whose degree is more than $O(d/t)$. The contribution from the children of these vertices to $N_+(v)$ might be too large. Hence we need to argue that not too many vertices of this type can be embedded in M_v.

First, observe that using Lemma 3.3 and the definition of M_v, the size of M_v is bounded by

$$|M_v| \le \frac{e(M_v, N_+(v))}{2s^{\frac{1}{t-1}} d^{\frac{t-2}{t-1}}} \le \frac{2td}{2s^{\frac{1}{t-1}} d^{\frac{t-2}{t-1}}} \le td^{\frac{1}{t-1}}.$$

Similarly, if we denote by Q the vertices of T with degrees at least $\frac{1}{64t}d$, the number of such vertices is bounded by

$$|Q| \le \frac{2|T|}{\frac{1}{64t}d} \le \frac{\frac{1}{32} d^{\frac{t}{t-1}}}{\frac{1}{64t}d} = 2td^{\frac{1}{t-1}}.$$

Our goal is to prove that not many vertices from Q can be embedded in M_v. For that purpose, we also need to define a new type of "bad event" $\mathcal{C}_v$ and "dangerous event" $\mathcal{E}_v$.

The event $\mathcal{E}_v$ occurs if any vertex of the tree is embedded in M_v. The event $\mathcal{C}_v$ occurs if after the first vertex embedded in M_v, at least 8 vertices from Q are embedded in M_v.

Now, consider any tree vertex $q \in Q$. At the moment when we embed q, there are at least $\frac{1}{4}d$ choices, unless $\mathcal{B}_w$ happened for some vertex w and the algorithm has failed already. Since $|M_v| \le td^{\frac{1}{t-1}}$, the probability of embedding q into M_v, even conditioned on any previous history $\mathcal{H}'$, is

$$\mathbb{P}[f(q) \in M_v \mid \mathcal{H}'] \le \frac{|M_v|}{\frac{1}{4}d} \le \frac{4td^{\frac{1}{t-1}}}{d} \le \frac{4t}{d^{3/4}}$$

for $t \geq 5$. We condition on any history $\mathcal{H}$ up to the first vertex embedded in M_v, and estimate the probability that at least 8 vertices from Q are embedded in M_v after this moment. For any particular 8-tuple from Q, this probability is bounded by $(4t/d^{3/4})^8 = (4t)^8/d^6$. The number of possible 8-tuples in Q is at most $|Q|^8 \leq (2td^{1/(t-1)})^8 \leq (2t)^8 d^2$ for $t \geq 5$. Hence,

$$\mathbb{P}[\mathcal{C}_v \mid \mathcal{H}] \leq \frac{(4t)^8}{d^6}(2t)^8 d^2 = \frac{8^8 t^{16}}{d^4}.$$

By averaging over all histories up to the moment when the first vertex is embedded in M_v, we get $\mathbb{P}[\mathcal{C}_v \mid \mathcal{E}_v] \leq 8^8 t^{16}/d^4$.

Consider the number of events $\mathcal{E}_v$ that can ever happen. For any event $\mathcal{E}_v$, there is a witness vertex $x \in V(T)$ mapped to $f(x) = w \in M_v$. Observe that the definition of $w \in M_v$ is symmetric with respect to (v, w), i.e., we also have $v \in M_w$. We know that $|M_w| \leq td^{1/(t-1)}$ for any $w \in V$, therefore each vertex of the tree can witness at most $td^{1/(t-1)}$ events $\mathcal{E}_v$. In total, we can have at most $|T| \cdot td^{1/(t-1)} \leq d^{t/(t-1)} \cdot td^{1/(t-1)} \leq td^2$ events $\mathcal{E}_v$. Since $\mathcal{E}_v$ can occur for at most td^2 vertices in any given run of the algorithm, we have $\sum_{v \in V} \mathbb{P}[\mathcal{E}_v] \leq td^2$. Hence,

$$\begin{aligned}
\mathbb{P}[\exists v \in V; \mathcal{C}_v \text{ occurs}] &\leq \sum_{v \in V} \mathbb{P}[\mathcal{C}_v] = \sum_{v \in V} \mathbb{P}[\mathcal{C}_v \mid \mathcal{E}_v]\mathbb{P}[\mathcal{E}_v] \\
&\leq \frac{8^8 t^{16}}{d^4} \sum_{v \in V} \mathbb{P}[\mathcal{E}_v] \leq \frac{8^8 t^{16}}{d^4} td^2 \leq \frac{8^8 t^{17}}{d^2}
\end{aligned}$$

which tends to 0 for $d \to \infty$. So, with high probability, none of the events $\mathcal{C}_v$ happen.

Given that $\mathcal{C}_v$ does not occur for any vertex, we can carry out the same analysis we used to prove Theorem 3.1. The only difference is that each vertex v might have up to 9 vertices from Q embedded in M_v (8 plus the first vertex ever embedded in M_v). Since the degrees in T are bounded by $\frac{1}{256}d$, even if the children of these vertices were embedded arbitrarily, they still can only occupy at most $\frac{9}{256}d$ vertices in $N_+(v)$. The number of vertices in $N_+(v)$ occupied through vertices in L_v, as well as the contribution of the children of vertices in T with degree $O(d/t)$ that were embedded in M_v, can be analyzed just like in Theorem 3.1. Thus, with high probability, at most $\frac{1}{2}d + \frac{9}{256}d + 2t < \frac{3}{4}d$ vertices are occupied in any neighborhood, so at least $\frac{1}{4}d$ vertices are always available to embed any vertex of the tree. $\square$

4 Graphs of fixed girth

In this section we consider the problem of embedding trees into graphs which have no cycle of length less than $2k + 1$ for some $k > 1$. (If the shortest cycle in a graph has length $2k + 1$, such a graph is said to have *girth* $2k + 1$.) We also assume that the minimum degree in our graph is at least d. It is easy to see that any such graph G must have $\Omega(d^k)$ vertices, because up to distance k from any vertex v, G locally looks like a tree. It is widely believed that graphs of minimum degree d, girth $2k+1$, and order $O(d^k)$ do exist for all fixed k and large d. Such constructions are known when $k = 2, 3$ and 5. Since our graph might have order $O(d^k)$, we cannot aspire to embed trees of order larger than $O(d^k)$ in G. This is what we achieve. For the purpose of analysis, we need to slightly modify our previous algorithms.

Algorithm 3. *For each $v \in V$, fix a set of d neighbors $N_+(v) \subseteq N(v)$. Assume that T is a rooted tree with root r. Start by making k random moves from an arbitrary vertex $v_1 \in V$, in each step choosing a random neighbor $v_{i+1} \in N_+(v_i)$. Embed the root of the tree at $f(r) = v_k$.*

As long as T is not completely embedded, take an arbitrary vertex $s \in V(T)$ which is embedded but whose children are not. If $f(s)$ has enough available neighbors in $N_+(f(s))$ unoccupied by other vertices of T, embed the children of s among these vertices uniformly at random. Otherwise, fail.

The following is our main result for graphs of girth $2k + 1$.

Theorem 4.1 *Let G be a graph with minimum degree d and girth $2k+1$. Then for any constant $\epsilon \leq \frac{1}{2k}$, Algorithm 3 succeeds with high probability in embedding any tree T of order $\frac{1}{4}\epsilon d^k$ and maximum degree $\Delta(T) \leq d - 2\epsilon d - 2$.*

To prove this theorem, we will generalize the analysis of the C_4-free case to allow the embedding of substantially larger trees. The solution is to consider multiple levels of neighborhoods for each vertex. Starting from any vertex $v \in V(G)$, we have the property that up to distance k from v, G looks like a tree (otherwise we get a cycle of length at most $2k$). Consequently, for any vertex w, there can be at most one path of length k from w to v . Therefore, embedding a subtree whose root is placed at w cannot impact the neighborhood of v too much.

In fact, neighbors to be used in the embedding are chosen only from a subset of d neighbors $N_+(v)$. We can define an orientation of G where each vertex has out-degree exactly d, by orienting all edges from v to $N_+(v)$. (Some edges can be oriented both ways.) Then, branches of the tree T are embedded along *directed paths* in G.

Definition 4.2 *For a rooted tree T, with a natural top-to-bottom orientation, let $L_i(x)$ define the set of descendants i levels down from $x \in V(T)$.*

For a tree vertex $x \in V(T)$, denote by $X_{v,x}$ the number of vertices in $L_{k-1}(x)$ that end up embedded in $N_+(v)$ before the children of $f^{-1}(v)$ are embedded.

For a vertex $v \in V(G)$, denote by X_v the total number of vertices in T that end up embedded in $N_+(v)$ before the children of $f^{-1}(v)$ are embedded.

We extend T to a larger rooted tree T^* by adding a path of length $k - 1$ above the root of T, and making the endpoint of this path the root of T^*. Observe that our embedding algorithm essentially proceeds as if it is embedding T^*, except the first $k - 1$ steps do not occupy any vertices of G. Each embedded vertex $y \in V(T)$ is a $(k - 1)$-descendant of some $x \in V(T^*)$ and hence $V(T) = \bigcup_{x \in V(T^*)} L_{k-1}(x)$. By summing up the contributions over $x \in V(T^*)$, we get

$$X_v = \sum_{x \in V(T^*)} X_{v,x}.$$

Our goal is to apply tail estimates on X_v in order to bound the probabilities of "bad events". Just like before, we need to be careful in summing up these probabilities, since the order of the graph might be too large for a union bound. We start "watching out" for the bad event $\mathcal{B}_v$ only after a "dangerous event" $\mathcal{D}_v$ occurs. We also stop our algorithm immediately after the first bad event happens.

Event $\mathcal{B}_v$ occurs when $X_v > 2\epsilon d + 2$. Event $\mathcal{D}_v$ *occurs whenever at least two vertices in $N_+(v)$ can be reached by directed paths of length at most $k-1$, avoiding v, from the embedding of T^*. By the embedding of T^*, we also mean the vertices visited in the first $k-1$ steps of the algorithm, which are not really occupied.*

Suppose q_1, q_2 are the first two vertices in $N_+(v)$ that can be reached by directed paths of length at most $k-1$, avoiding v, from the embedding of T^*. Then we define a modified random variable $\tilde{X}_{v,x}$ as the number of vertices in $L_{k-1}(x)$ which are embedded in $N_+(v) \setminus \{q_1, q_2\}$, but not through v itself. In other words, these random variables count the vertices occupied in $N_+(v)$, except for q_1 and q_2. Observe that $X_v \leq \sum_{x \in V(T^*)} \tilde{X}_{v,x} + 2$.

Lemma 4.3 *Assume the girth of G is at least $2k+1$. Fix an ordering of the vertices of T^* starting from the root, $(x_1, x_2, x_3, \ldots)$, as they are processed by the algorithm. Let $\mathcal{H}$ be a fixed history of running the algorithm until two vertices $q_1, q_2 \in N_+(v)$ can be reached from an embedded vertex by a directed path (avoiding v) of length at most $k-1$. Then for any vertex $x_i \in V(T^*)$, $\tilde{X}_{v,x_i}$ is a 0/1 random variable such that*

$$\mathbb{P}[\tilde{X}_{v,x_i} = 1 \mid \mathcal{H}, \tilde{X}_{v,x_1}, \tilde{X}_{v,x_2}, \ldots, \tilde{X}_{v,x_{i-1}}] \leq \frac{|L_{k-1}(x_i)|}{(d - 2\epsilon d - 2)^{k-1}}.$$

Proof. First, note that any vertex x_i embedded during the history $\mathcal{H}$ has $\tilde{X}_{v,x_i} = 0$. (The only vertices in $N_+(v)$ possibly reachable within $k-1$ steps from $f(x_i)$ are q_1 and q_2.) Therefore we can assume that the embedding of x_i, together with the embedding of the subtree of its descendants in T^*, is still undecided at the end of $\mathcal{H}$. Let $\mathcal{K}$ denote the event that x_i is embedded such that there is a directed path of length exactly $k-1$ from $f(x_i)$ to $N_+(v)$, which avoids v and has an endpoint in $N_+(v)$ other than q_1, q_2. Observe that this is the only way $\tilde{X}_{v,x_i}$ could be non-zero. Indeed, if $\tilde{X}_{v,x_i} = 1$, then there is a branch of the tree T^* of length $k-1$ from x_i to some y, which was mapped to a path from $f(x_i)$ to $N_+(v)$ such that the next-to-last vertex is not v. However, such a path from $f(x_i)$ to $N_+(v)$, if it exists, is unique. If we had two different paths like this, we could extend them to two paths of length k between $f(x_i)$ and v, contradicting the girth assumption. Note that $\mathcal{K}$ occurs only if this unique path leads to a vertex of $N_+(v)$ other than q_1 or q_2. Also, we have that at most one vertex $y \in L_{k-1}(x_i)$ can be embedded in $N_+(v)$. The variable $\tilde{X}_{v,x_i}$ is equal to 1 when this happens for some $y \in L_{k-1}(x_i)$, and 0 otherwise.

We bound the probability that $\tilde{X}_{v,x_i} = 1$, conditioned on $(\mathcal{H}, \tilde{X}_{v,x_1}, \ldots, \tilde{X}_{v,x_{i-1}})$. In fact, we condition even more strongly on a fixed embedding $\mathcal{E}$ of all vertices of T except for the descendants of x_i. We also assume that $\mathcal{E}$ satisfies $\mathcal{K}$, i.e., $f(x_i)$ is at distance exactly $k-1$ from $N_+(v)$, since otherwise $\tilde{X}_{v,x_i} = 0$. We claim that any such embedding determines the values of $\tilde{X}_{v,x_1}, \ldots, \tilde{X}_{v,x_{i-1}}$. For vertices x_j such that $L_{k-1}(x_j)$ does not intersect the subtree of x_i, this is clear because the embedding of these vertices is fixed. However, even if $L_{k-1}(x_j)$ intersects the subtree of x_i, $\tilde{X}_{v,x_j}$ is still determined, since none of these vertices can be embedded into $N_+(v)$. Indeed, any descendant of x_i which is in $L_{k-1}(x_j)$ must also be in $L_{k'}(x_i)$ for some $k' < k-1$. If the embedding of $L_{k'}(x_i)$ intersects $N_+(v)$, we obtain that there are two paths from $f(x_i)$ to v, one of length k and another of length $k'+1 < k$. Together, they form a cycle of length shorter than the girth, a contradiction.

Now fix a vertex $y \in L_{k-1}(x_i)$. Each vertex $x_j \in T^*$, when embedded, chooses randomly from one of the available neighbors of the vertex of G at which its parent was embedded. As long as no bad event has happened so far (otherwise the algorithm already failed), there are at least $d-2\epsilon d-2$ candidates available for $f(x_j)$. Therefore, each particular vertex has probability at most $1/(d-2\epsilon d-2)$ of being chosen to be $f(x_j)$. The probability that $f(y) \in N_+(v)$ is the probability that our embedding follows a particular path of length $k-1$. By the above discussion, this probability is at most $1/(d-2\epsilon d-2)^{k-1}$. (Note that by our conditioning, this path might be already blocked by the placement of other vertices; in such a case, the probability is actually 0.) Using the union bound, we have

$$\mathbb{P}[\tilde{X}_{v,x_i} = 1 \mid \mathcal{E}] \le \frac{|L_{k-1}(x_i)|}{(d-2\epsilon d-2)^{k-1}}.$$

Since the right hand side of this inequality is a constant, independent of the embedding, we get the same bound conditioned on $(\mathcal{H}, \tilde{X}_{v,x_1}, \ldots, \tilde{X}_{v,x_{i-1}}, \mathcal{K})$ and hence also conditioned on $(\mathcal{H}, \tilde{X}_{v,x_1}, \ldots, \tilde{X}_{v,x_{i-1}})$. □

Now we are ready to use our supermartingale tail estimate from Proposition 1.1 to bound the probability of a bad event.

Lemma 4.4 *Assume $\epsilon \le \frac{1}{2k}$ and $|T| \le \frac{1}{4}\epsilon d^k$. For any vertex $v \in V(G)$, condition on the dangerous event $\mathcal{D}_v$. Then for sufficiently large d, the probability that the bad event $\mathcal{B}_v$ happens is*

$$\mathbb{P}[\mathcal{B}_v \mid \mathcal{D}_v] \le e^{-\epsilon d/3}.$$

Proof. The bad event means that $X_v > 2\epsilon d + 2$. As before, we first condition on any history $\mathcal{H}$ up to the point when $\mathcal{D}_v$ happens. At this point, two vertices $q_1, q_2 \in N_+(v)$ are within distance $k-1$ of the embedding of T^* constructed so far. We consider these two vertices effectively occupied. Our goal is to prove that the number of additional occupied vertices in $N_+(v)$ is small, namely $\sum_{i=1}^{|T^*|} \tilde{X}_{v,x_i} \le 2\epsilon d$.

By Lemma 4.3, we know that

$$\mathbb{P}[\tilde{X}_{v,x_i} = 1 \mid \mathcal{H}, \tilde{X}_{v,x_1}, \ldots, \tilde{X}_{v,x_{i-1}}] \le \frac{|L_{k-1}(x_i)|}{(d-2\epsilon d-2)^{k-1}}.$$

Therefore the expectation of $\tilde{X}_v = \sum_{i=1}^{|T^*|} \tilde{X}_{v,x_i}$ is bounded by

$$\mathbb{E}[\tilde{X}_v] = \sum_{i=1}^{|T^*|} \mathbb{E}[\tilde{X}_{v,x_i}] \le \sum_{i=1}^{|T^*|} \frac{|L_{k-1}(x_i)|}{(d-2\epsilon d-2)^{k-1}} \le \frac{|T|}{(d-2\epsilon d-2)^{k-1}} < \frac{4|T|}{d^{k-1}} \le \epsilon d.$$

Here we used that $\epsilon \le \frac{1}{2k}$, d is large enough, and $|T| \le \frac{1}{4}\epsilon d^k$. So, we can set $\mu = \epsilon d$, $\delta = 1$, and use Proposition 1.1 to conclude that

$$\mathbb{P}\big[\tilde{X}_v > 2\epsilon d \mid \mathcal{H}\big] \le e^{-\epsilon d/3}.$$

The same holds when we condition on the event $\mathcal{D}_v$, which is the disjoint union of all such histories $\mathcal{H}$. Consequently, $X_v \le \tilde{X}_v + 2 \le 2\epsilon d + 2$ with high probability, which concludes the proof. □

To finish the proof of Theorem 4.1, we show that with high probability, $\mathcal{B}_v$ does not happen for any vertex $v \in V$. First, we examine how many events $\mathcal{D}_v$ can possibly occur for a given run of the algorithm. Every vertex v for which $\mathcal{D}_v$ happens has a "witness pair" of vertices in $N_+(v)$ satisfying the condition that they can be reached by directed paths of length at most $k-1$ from the embedding of T^*. The number of such vertices is at most $|T^*|d^{k-1} \le d^{2k}$. Also, observe that the same pair can witness at most 1 event $\mathcal{D}_v$, or else we have a 4-cycle in G which contradicts the high girth property. Hence the number of possible witness pairs is at most

$$\binom{d^{2k}}{2} \leq d^{4k},$$

and each event $\mathcal{D}_v$ has a unique witness pair. Therefore, the expected number of events $\mathcal{D}_v$ is

$$\sum_v \mathbb{P}[\mathcal{D}_v] \leq d^{4k}.$$

Now we bound the probability that any bad event $\mathcal{B}_v$ occurs.

$$\begin{aligned} \mathbb{P}[\exists v \in V; \mathcal{B}_v \text{ occurs}] &\leq \sum_{v \in V} \mathbb{P}[\mathcal{B}_v] = \sum_{v \in V} \mathbb{P}[\mathcal{B}_v \mid \mathcal{D}_v]\mathbb{P}[\mathcal{D}_v] \\ &\leq e^{-\epsilon d/3} \sum_{v \in V} \mathbb{P}[\mathcal{D}_v] \leq d^{4k} e^{-\epsilon d/3}. \end{aligned}$$

For a constant k and $d \to \infty$, this probability tends to 0. □

5 Concluding remarks

In this paper we have shown that a very simple randomized algorithm can efficiently find tree embeddings with near-optimal parameters, surpassing some previous results achieved by more involved approaches. Here are a few natural questions which remain open.

- It would be interesting to extend our results from graphs of girth $2k+1$ to graphs without cycles of length $2k$. For $k = 3$, this follows from our work combined with a result of Györi. In [13] he proved that every bipartite C_6-free graph can be made also C_4-free by deleting at most half of its edges. Therefore given a C_6-free graph with minimum degree d, we can first take its maximum bipartite subgraph. This will decrease the number of edges by at most factor of two. Then we can use the above mentioned result of Györi to obtain a C_4-free and C_6-free graph which has at least a quarter of the original edges, i.e., average degree at least $d/4$. In this graph we can find a subgraph where the minimum degree is at least $d/8$ (1/2 of the average degree). Since it is bipartite, this subgraph has no cycles of length shorter than 7. This shows that every C_6-free graph G with minimum degree d contains a subgraph G' of girth at least 7 whose minimum degree is a constant fraction of d. Using our result we can embed in G' (and hence also in G) every tree of order $O(d^3)$ and minimum degree $O(d)$.

  More generally, it is proved in [19] that any C_{2k}-free graph contains a C_4-free subgraph with at least $\frac{1}{2(k-1)}$-fraction of its original edges. Moreover, it is conjectured in [19] that any C_{2k}-free graph contains a subgraph of girth $2k+1$ with at least an ϵ_k-fraction of the edges. If this conjecture is true it shows that the tree embedding problems for C_{2k}-free graphs and graphs of girth $2k+1$ are equivalent up to constant factors.

- Our algorithm also works well for random graphs $G_{n,p}$, although we can only prove that it achieves nearly-optimal results for a certain range of p. Namely, when $p = n^{a-1}$ for some constant $a > 0$, we can show that our algorithm embeds with high probability any tree with $O(an)$ vertices and degrees bounded by $O(apn)$. This is somewhat weaker than the results on random graphs that we mentioned in the introduction. Still, it might be that our algorithm performs better than our analysis shows.

- Finally we wonder whether there are any additional interesting families of graphs for which our simple randomized algorithm succeeds in embedding trees with nearly optimal parameters. Although we believe that our algorithm is nearly optimal in a wide range of settings, it would be also interesting to find examples showing where our algorithm breaks down.

Acknowledgment. We would like to thank an anonymous referee for bringing reference [14] to our attention and for helpful comments which significantly improved the presentation of this paper.

References

[1] M. Ajtai, J. Komlós, M. Simonovits and E. Szemerédi, The exact solution of the Erdős-T. Sós conjecture for (large) trees, in preparation.

[2] N. Alon, M. Krivelevich and B. Sudakov, Embedding nearly-spanning bounded degree trees, *Combinatorica* 27 (2007), 629–644.

[3] N. Alon, L. Rónyai and T. Szabó, Norm-graphs: variations and applications, *J. Combinatorial Theory Ser. B* 76 (1999), 280–290.

[4] I. Benjamini and Y. Peres, Markov chains indexed by trees, *Ann. Probability* 22 (1994), 219–243.

[5] I. Benjamini and O. Schramm, Every Graph with a Positive Cheeger Constant Contains a Tree with a Positive Cheeger Constant, *Geometric and Functional Analysis* 7 (1997), 403–419.

[6] C.T. Benson, Minimal regular graphs of girth eight and twelve, *Canad. J. Math.* 18 (1966), 1091–1094.

[7] S.N. Bhatt, F. Chung, F.T. Leighton and A. Rosenberg, Universal graphs for bounded-degree trees and planar graphs, *SIAM J. Discrete Math.* 2 (1989), 145–155.

[8] S. Brandt and E. Dobson, The Erdős-Sós conjecture for graphs of girth 5, *Discrete Math.* 150 (1996), 411–414.

[9] P. Erdős, Z. Füredi, M. Loebl and V. Sós, Discrepancy of Trees, *Studia Sci. Math. Hungarica* 30 (1995), 47–57.

[10] P. Erdős and A. Rényi, On a problem in the theory of graphs (in Hungarian), *Publ. Math. Inst. Hungar. Acad. Sci.* **7** (1962), 215–235.

[11] J. Friedman and N. Pippenger, Expanding graphs contain all small trees, *Combinatorica* 7:1 (1987), 71–76.

[12] T. Jiang, On a conjecture about trees in graphs with large girth, *J. of Combinatorial Theory B*, 83:2 (2001), 221–232.

[13] E. Györi, C_6-free bipartite graphs and product representation of squares, *Discrete Math.* 165/166 (1997), 371–375.

[14] P. Haxell and Y. Kohayakawa, The size-Ramsey number of trees, *Israel J. Math.* 89:1-3 (1995), 261–274.

[15] P. Haxell and T. Łuczak, Embedding trees into graphs of large girth, *Discrete Math.* 216 (2000), 273–278.

[16] P. Haxell, Tree embeddings, *J. Graph Theory* 36 (2001), 121–130.

[17] J. Kollár, L. Rónyai and T. Szabó, Norm-graphs and bipartite Turan numbers, *Combinatorica* 16 (1996), 399–406.

[18] T. Kövari, V.T. Sós and P. Turán, On a problem of K. Zarankiewicz,*Colloquium Math.* 3 (1954), 50–57.

[19] D. Kühn and D. Osthus, 4-cycles in graphs without a given even cycle, *Journal of Graph Theory* 48 (2005), 147–156.

[20] M. Krivelevich and B. Sudakov, Pseudo-random graphs, in *More Sets, Graphs and Numbers*, Bolyai Society Mathematical Studies 15, Springer, 2006, 199–262.

[21] M. Krivelevich and B. Sudakov, Minors in expanding graphs, *Geometric and Functional Analysis* 19 (2009), 294–331.

[22] N. Madras and G. Slade, The Self-Avoiding Walk, *Birkhauser, Boston* (1993).

[23] C. McDiarmid, Concentration, in: *Probabilistic methods for algorithmic discrete mathematics*, Algorithms and Combinatorics 16, Springer, Berlin, 1998, 195–248.

[24] A. Naor and J. Verstraete, A note on bipartite graphs without $2k$-cycles, *Combinatorics, Probability and Computing* 14:5-6 (November 2005), 845–849.

[25] L. Pósa, Hamiltonian circuits in random graphs, *Discrete Math.* 14 (1976), 359–364.

[26] B. Sudakov and J. Verstraete, Cycles in sparse graphs, *Combinatorica* 28 (2008), 357–372.

[27] W. F. de la Vega, Long paths in random graphs, *Studia Sci. Math. Hungar.* 14 (1979), 335–340.

[28] W. F. de la Vega, Trees in sparse random graphs, *J. Combin. Theory B* 45 (1988), 77–85.

[29] Y. Zhao. Proof of the $(n/2 - n/2 - n/2)$ conjecture for large n, *manuscript* (2007).