Discrete Applied Mathematics 292 (2021) 64–71

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Note

A note on the independence number, domination number and related parameters of random binary search trees and random recursive trees

Michael Fuchs [a,1], Cecilia Holmgren [b,2,*], Dieter Mitsche [c,3], Ralph Neininger [d]

[a] Department of Mathematical Sciences, National Chengchi University, Taiwan
[b] Department of Mathematics, Uppsala University, Sweden
[c] Institut Camille Jordan (UMR 5208), Univ. de Lyon, Univ. Jean Monnet, France
[d] Institute for Mathematics, Goethe University Frankfurt, Germany

ARTICLE INFO

Article history:
Received 11 February 2020
Received in revised form 27 November 2020
Accepted 8 December 2020
Available online 14 January 2021

Keywords:
Independence number
Domination number
Clique cover number
Random recursive trees
Random binary search trees
Fringe trees
Central limit laws

ABSTRACT

We identify the mean growth of the independence number of random binary search trees and random recursive trees and show normal fluctuations around their means. Similarly we also show normal limit laws for the domination number and variations of it for these two cases of random tree models. Our results are an application of a recent general theorem of Holmgren and Janson on fringe trees in these two random tree models.

1. Introduction and results

In this note we study the independence number, the domination number and related parameters of random binary search trees and random recursive trees asymptotically. First, in Section 2, we derive asymptotics for the mean and variance and provide central limit laws for the independence number of both tree models. This covers a few other graph parameters which are affine functions of the independence number, see Remark 1.7(c). In Section 3, we also provide central limit laws for the domination number and related parameters for both of these cases of random tree models. Finally, albeit coinciding with the independence number on trees, we also give a direct proof of such a theorem for the clique cover number in Section 4.

* Corresponding author.
E-mail address: cecilia.holmgren@math.uu.se (C. Holmgren).
1 The work by Michael Fuchs was partially supported by Ministry of Science and Technology (MOST), Taiwan MOST-109-2115-M-004-003-MY2
2 The work of Cecilia Holmgren is supported by the Swedish Research Council, the Ragnar Söderberg Foundation, Sweden and the Knut and Alice Wallenberg Foundation, Sweden.
3 Dieter Mitsche has been supported by IDEXLYON of Université de Lyon (Programme Investissements d'Avenir), France ANR16-IDEX-0005 and ANR, (AAPG, GrHyDy), France ANR-20-CE40-000.

https://doi.org/10.1016/j.dam.2020.12.013
0166-218X/© 2020 Published by Elsevier B.V.

We first recall the parameters under consideration and present the models of trees we are looking at and state our results.

Independence and domination number. The *independence number* of a graph is the size of a maximum independent set in the graph, where an independent set is a subset of the vertices of the graph so that no two vertices of this subset are connected (are neighbors) within the graph. The independence number is an important and well-known graph parameter: besides its applications in scheduling theory, coding theory and collusion detection in voting pools (see [1,5,21]), it has attracted a lot of interest especially in theoretical computer science: the independence number is well known to be NP-hard to compute in general, see for example [31]. Since then, exact fast exponential algorithms have been developed (see [27,28]) as well as polynomial-time algorithms for special graph classes (claw-free graphs, P_5-free graphs, perfect graphs, see [15,24,29]). In general, it is also NP-hard to approximate the independence number (that is, it is not possible to approximate it up to a constant factor in polynomial time) [3], but again for special graph classes such as planar graphs, or more generally, for graphs closed under taking minors, polynomial-time approximation schemes do exist [2,14]. For bipartite graphs, thus in particular trees, by König's theorem, all vertices not in the minimum vertex cover can be included in a maximum independent set (see also the remark below), and thus the independence number can be found in polynomial time. In combinatorics, it has also received considerable attraction, starting with the early work by Bollobás [4].

Given a finite graph G with vertex set V, a subset $W \subset V$ is called a dominating set for V if every vertex in V lies at graph distance at most 1 from W. The *domination number* of G is then defined to be the minimum number m such that there exists a dominating set W of size m. Finding dominating sets is important in finding 'central' or 'important' sets of vertices in a network, in contexts such as facility location [16], molecular biology [26] and in wireless networks [33]. Dominating sets have attracted considerable attention in discrete mathematics (see [16,17] and [18]) and as in the case of the independence number, in theoretical computer science: it was shown already in the 1970s (see [22]) that the domination number is NP-hard to compute, and it is also NP-hard to approximate up to a logarithmic factor in general [30]. Since then, as in the case of the independence number, exact fast exponential algorithms have been developed [12,35], and faster algorithms for special graph classes have been found as well (see for example [34] for series–parallel graphs). For trees, linear-time algorithms are known [7].

Random recursive tree and random binary search tree. A *random recursive tree* is a labeled rooted tree which can be constructed as follows. For the first step we start with the root vertex labeled 1. In the nth step, $n \geq 2$, one of the existing vertices labeled $1, \dots, n-1$ is chosen uniformly at random where a vertex with label n is attached. Subsequently, a random recursive tree with n vertices is denoted by Λ_n. For reference, see the survey of Smythe and Mahmoud [32]. We will need the following fact: A random recursive tree with n vertices can be cut into two trees by removing the edge between the root vertex labeled 1 and the vertex labeled 2. This yields two trees both with a random size, both sizes being uniformly distributed on $\{1, \dots, n-1\}$. Moreover, conditional on their sizes, these two trees are independent and both are (after proper relabeling of their vertices) random recursive trees of their respective size.

The *random binary search tree* can be constructed from a uniformly distributed random permutation $(\Pi_1, \dots, \Pi_n)$ of $\{1, \dots, n\}$. The first number Π_1 becomes the root of the tree. Then the numbers $\Pi_2, \dots, \Pi_n$ are successively inserted recursively. Each number is compared with the root. If it is smaller than the root, it is directed to the root's left subtree, otherwise to its right subtree. There, this procedure is recursively iterated until an empty subtree is reached, where the number is inserted as a new vertex. Subsequently, a random binary search tree with n vertices is denoted by $\mathcal{T}_n$. For reference, see Knuth [23]. We need the following decomposition property: The left and right subtrees at the root of the binary search tree both have random sizes uniformly distributed on $\{0, \dots, n-1\}$. Conditional on their sizes they are independent and both are (after proper relabeling of their vertices) random binary search trees of their respective sizes.

Results on the independence number. We denote by I_n the independence number of $\mathcal{T}_n$ and by $\widehat{I_n}$ the independence number of Λ_n. We have the following asymptotic results:

Theorem 1.1. *For the independence number I_n of a random binary search tree with n vertices we have, as $n \to \infty$, that $\mathbb{E}[I_n] = \mu n + O(1)$, $\mathrm{Var}(I_n) \sim \sigma^2 n$ and*

$$\frac{I_n - \mu n}{\sqrt{n}} \xrightarrow{d} \mathcal{N}(0, \sigma^2)$$

with

$$\mu = 2(\sqrt{5} - 3) \int_0^1 \frac{x^{\sqrt{5}} - 1}{(3\sqrt{5} - 7)x^{\sqrt{5}} + 2} \mathrm{d}x = 0.54287631\dots \tag{1}$$

and a constant $\sigma > 0$.

Theorem 1.2. *For the independence number $\widehat{I_n}$ of a random recursive tree with n vertices we have, as $n \to \infty$, that $\mathbb{E}[\widehat{I_n}] = \widehat{\mu} n + O(1)$, $\mathrm{Var}(\widehat{I_n}) \sim \widehat{\sigma}^2 n$ and*

$$\frac{\widehat{I_n} - \widehat{\mu} n}{\sqrt{n}} \xrightarrow{d} \mathcal{N}(0, \widehat{\sigma}^2)$$

with the Euler–Gompertz constant

$$\widehat{\mu} = \int_0^1 \frac{1}{1-\log x}\,\mathrm{d}x = 0.59634736\ldots \tag{2}$$

and a constant $\widehat{\sigma} > 0$.

Remark 1.3. Stephan Wagner (Stellenbosch University) informed us that he and his student Kenneth Dadedzi have an independent approach to results similar of our Theorems 1.1 and 1.2; they use generating functions to determine the spectrum of the Laplacian operator on these trees, see [9]. Stephan also informed us that our representation (7) for $\widehat{\mu}$ has the explicit integral representation given in (2); see also [25].

Results on the domination number. For the domination number of random binary search trees and random recursive trees we have similar results.

Theorem 1.4. *For the domination number D_n of a random binary search tree with n vertices we have, as $n \to \infty$, that $\mathbb{E}[D_n] = \nu n + O(1)$, $\mathrm{Var}(D_n) \sim \tau^2 n$ with some constants $\nu, \tau > 0$ and*

$$\frac{D_n - \nu n}{\sqrt{n}} \xrightarrow{d} \mathcal{N}(0, \tau^2).$$

Similarly, for the domination number $\widehat{D}_n$ of a random recursive tree with n vertices we have, as $n \to \infty$, that $\mathbb{E}[\widehat{D}_n] = \widehat{\nu} n + O(1)$, $\mathrm{Var}(D_n) \sim \widehat{\tau}^2 n$ with some constants $\widehat{\nu}, \widehat{\tau} > 0$ and

$$\frac{\widehat{D}_n - \widehat{\nu} n}{\sqrt{n}} \xrightarrow{d} \mathcal{N}(0, \widehat{\tau}^2).$$

Remark 1.5. For the case of the random recursive tree, it has been shown in [8] that the numerical value of the constant satisfies $\widehat{\nu} = 0.3745\ldots$.

Remark 1.6. A variation of the domination number, the so-called *k-domination number* of a graph, was introduced in [11]. This is defined as the minimum size of a set S of vertices in a graph such that each vertex of the graph (outside the set S) has at least k neighbors in S. We can analyze these numbers as well in the case of random binary search trees and random recursive trees and obtain normal limit laws corresponding to the ones in Theorem 1.4. For binary search trees, where each vertex has degree at most 3, however, we also have to assume that $k \leq 3$ (to avoid the trivial case $|S| = n$), while for random recursive trees, we may consider the k-domination number for any constant $k > 0$.

Remark 1.7. (a) Various quantities for random binary search trees have systematically been studied with respect to limit distributions by Devroye [10] and Hwang and Neininger [20]. However, the independence number and the domination number do not fit under the assumptions made in those two studies. Our proof relies on a recent refined study of fringe trees of random binary search trees and random recursive trees of Holmgren and Janson [19] which extends parts of the results of [10,20].

(b) Holmgren and Janson [19] also give general formulae for expectation and variance which cover our variances σ^2 and $\widehat{\sigma}^2$ in Theorems 1.1 and 1.2 and also expectations ν and $\widehat{\nu}$ as well as variances τ^2 and $\widehat{\tau}^2$ in Theorem 1.4. Their representations allow one to compute numerical approximations for these values.

(c) There are a few (other) related graph parameters which are covered by our results, since they are affine functions of the independence number: The *matching number* (also known as edge independence number) is the size of a maximum set of edges so that no two edges have a common vertex. For all bipartite graphs and in particular trees, the matching number and the independence number add up to the size of the tree. Hence, for the matching numbers M_n and $\widehat{M}_n$ of a random binary search tree and a random recursive tree with n vertices respectively, we have $\mathbb{E}[M_n] = (1-\mu)n + O(1)$ with the same variance and limit as for I_n in Theorem 1.1, and $\mathbb{E}[\widehat{M}_n] = (1-\widehat{\mu})n + O(1)$ with the same variance and limit as for $\widehat{I}_n$ in Theorem 1.2.

The *edge cover number* of a connected graph is the minimum number of edges so that all vertices are incident to at least one edge. The edge cover number and the independence number coincide for trees.

The *vertex cover number* is the minimum number of vertices such that every edge has at least one of these vertices as an endpoint. The matching number and the vertex cover number coincide for trees.

The multiplicity of the eigenvalue 1 of the *normalized Laplacian operator* of a tree is twice the independence number of the tree minus its size, see [6, Theorem 1]. Hence, Theorems 1.1 and 1.2 imply the asymptotics of this multiplicity of the two random tree models considered in the present note as well. See [9] for a more general study of the asymptotics of the spectra of these random trees.

The *clique cover number* of a finite graph G is the minimum number of colors needed to color properly the vertices of the complement of G (the complement of G has the same vertex set as G, and two vertices are adjacent in the complement of G if and only if they are not adjacent in G). For trees, the clique cover number coincides with the independence number. We give a variant of the derivation of Theorems 1.1 and 1.2 in terms of the clique cover number, see Section 4.

2. Independence number

For our proof we use a simple construction of a maximum independent set by starting at the leaves. For a rooted tree T (or a forest of rooted trees) denote by leaf(T) the set of leaves of T and by $p(\text{leaf}(T))$ the set of the parents of the leaves of T. Recursively, define

$$T^{[0]} := T \quad \text{and} \quad T^{[\ell]} := T^{[\ell-1]} \setminus \big(\text{leaf}(T^{[\ell-1]}) \cup p(\text{leaf}(T^{[\ell-1]}))\big) \text{ for } \ell \geq 1.$$

So, $T^{[0]}, T^{[1]}, T^{[2]}, \ldots$ is a sequence of rooted trees or forests of rooted trees starting with T where in each step all the leaves together with their parents are removed from the present tree or forest until we reach the empty graph. Note, that when starting with a tree the sequence generated may also contain forests.

Lemma 2.1. *Let T be a rooted tree. Then*

$$\bigcup_{\ell=0}^{\infty} \text{leaf}\left(T^{[\ell]}\right)$$

is a maximum independent set of T.

Proof. Let T be a rooted tree or forest of rooted trees. We first show that there is always a maximum independent set of T which contains leaf(T). To see this choose an arbitrary maximum independent set A of T. If A does not contain a leaf ν then it has to contain its parent $p(\nu)$. However, then also $(A \setminus \{p(\nu)\}) \cup \{\nu\}$ is a maximum independent set of T which now contains the leaf ν. Iterating this process implies the existence of a maximum independent set of T containing leaf(T).

Further, a maximum independent set containing leaf(T) cannot contain any vertex of $p(\text{leaf}(T))$ and hence consists of the union of leaf$(T^{[0]})$ and a maximum independent set of $T^{[1]}$. Applying the previous argument to $T^{[1]}$ and using induction implies the assertion. □

Subsequently, we call the maximum independent set of a rooted tree constructed in Lemma 2.1 the *layered independent set*.

A result of Holmgren and Janson [19]. Recalling notions from Holmgren and Janson [19] a *functional* of trees is a real-valued function of trees. For a rooted tree T and a vertex $v \in T$ the *fringe tree* $T(v)$ is the subtree rooted at $v \in T$ which consists of all descendants of v in T. For a functional f of rooted trees we define

$$F(T) = F(T; f) := \sum_{v \in T} f(T(v)). \tag{3}$$

Corollary 1.15 in [19] states that for a functional f with the growth condition $f(T) = O(|T|^{\alpha})$ for some $\alpha < \frac{1}{2}$ and the random binary search tree $\mathcal{T}_n$ we have $\mathbb{E}[F(\mathcal{T}_n)] \sim \mu_F n$, $\text{Var}(F(\mathcal{T}_n)) \sim \sigma_F^2 n$ as $n \to \infty$, and that $F(\mathcal{T}_n)$, after normalization, is asymptotically normally distributed. The constant μ_F is given by

$$\mu_F = \sum_{k=1}^{\infty} \frac{2\,\mathbb{E}[f(\mathcal{T}_k)]}{(k+1)(k+2)}. \tag{4}$$

Note that in (1.25) in [19] also an expression for σ_F^2 is given. A similar result also holds for the random recursive tree Λ_n, where the corresponding constant $\widehat{\mu}_F$ is given by

$$\widehat{\mu}_F = \sum_{k=1}^{\infty} \frac{\mathbb{E}[f(\Lambda_k)]}{k(k+1)}. \tag{5}$$

Further note that the proofs in [19] also imply that $\mathbb{E}[F(\mathcal{T}_n)] = \mu_F n + O(1)$ and that $\mathbb{E}[F(\Lambda_n)] = \widehat{\mu}_F n + O(1)$ under the stronger growth assumption that $f(T) = O(1)$.

Putting things together now implies Theorems 1.1 and 1.2:

Proof of Theorems 1.1 and 1.2. Note that the independence number of a rooted tree can be covered as a function F in (3) as follows. We set f as the indicator function

$$f(T) := \begin{cases} 1, & \text{if the root of } T \text{ is contained in the layered independent set of } T, \\ 0, & \text{otherwise.} \end{cases}$$

The structure of the layered independent set in Lemma 2.1 implies that any vertex $v \in T$ is contained in the layered independent set of T if and only if it is contained in the layered independent set of $T(v)$.

Hence, the independence number of T is given by $F(T) = \sum_{v \in T} f(T(v))$ as in (3). This implies that $I_n = F(\mathcal{T}_n)$ and $\widehat{I}_n = F(\Lambda_n)$ in distribution. We have $f(T) = O(1)$. Hence, Corollary 1.15 of Holmgren and Janson [19] implies the assertions of Theorems 1.1 and 1.2 where $\sigma, \widehat{\sigma} > 0$ follows from numerical computation (see Remark 1.7(b)) and only the constants μ and $\widehat{\mu}$ need to be identified. In view of (4) and (5) we need to find $\mathbb{E}[f(\mathcal{T}_k)]$ and $\mathbb{E}[f(\Lambda_k)]$.

For the random recursive tree T note that T can be cut into two trees by removing the edge between the root vertex labeled 1 and the vertex labeled 2. We denote the two resulting trees by T_1 and T_2. Now, the root of T is contained in the layered independent set of T if and only if the root of T_1 is contained in the layered independent set of T_1 and the root of T_2 is not contained in the layered independent set of T_2. Now, the decomposition property of the random recursive tree mentioned in the introduction implies that with $\widehat{p}_n = \mathbb{E}[f(\Lambda_n)]$ we have the recurrence

$$\widehat{p}_n = \frac{1}{n-1}\sum_{j=1}^{n-1}(1-\widehat{p}_j)\widehat{p}_{n-j}, \quad n \geq 2, \tag{6}$$

with initial condition $\widehat{p}_1 := 1$. Furthermore, for the constant $\widehat{\mu}$ in Theorem 1.2 we have the representation

$$\widehat{\mu} = \sum_{k=1}^{\infty}\frac{\widehat{p}_k}{k(k+1)}. \tag{7}$$

Now, to find the integral expression for $\widehat{\mu}$ in (2) consider the generating function

$$\widehat{P}(z) := \sum_{k\geq 1}\widehat{p}_k z^k.$$

From (6) and the initial conditions we obtain

$$z\widehat{P}'(z) = -\widehat{P}(z)^2 + \frac{1}{1-z}\widehat{P}(z).$$

This Riccati equation can be solved by standard methods: We define $\widehat{Q}(z)$ as $\widehat{P}(z) = z\widehat{Q}'(z)/\widehat{Q}(z)$ and obtain

$$\widehat{Q}''(z) = \frac{1}{1-z}\widehat{Q}'(z),$$

which implies $\widehat{Q}'(z) = (1-z)^{-1}$ and thus $\widehat{Q}(z) = -\log(1-z) + c$ with a constant $c \in \mathbb{R}$. Hence, we obtain

$$\widehat{P}(z) = \frac{z}{(1-z)(-\log(1-z)+c)},$$

and the initial condition $\widehat{P}'(0) = 1$ yields $c = 1$. Now we obtain

$$\begin{aligned}\widehat{\mu} = \sum_{k=1}^{\infty}\frac{\widehat{p}_k}{k(k+1)} &= \int_0^1\int_0^t \frac{1}{(1-z)(1-\log(1-z))}\mathrm{d}z\mathrm{d}t \\ &= \int_0^1\int_z^1 \frac{1}{(1-z)(1-\log(1-z))}\mathrm{d}t\mathrm{d}z \\ &= \int_0^1 \frac{1}{(1-\log(1-z))}\mathrm{d}z,\end{aligned}$$

which, after substitution, is the expression in (2) for the Euler–Gompertz constant. This concludes the proof of Theorem 1.2.

For the binary search tree case note that the root of the tree T is contained in its layered independent set if and only if both children v_ℓ and v_r of the root are not contained in the layered independent set of $T(v_\ell)$ and $T(v_r)$ respectively. Now, the decomposition property of the random binary search tree mentioned in the introduction implies that with $p_k = \mathbb{E}[f(\mathcal{T}_k)]$ we have the relation

$$p_n := \frac{1}{n}\sum_{j=0}^{n-1}(1-p_j)(1-p_{n-1-j}), \quad n \geq 1, \tag{8}$$

with initial value $p_0 := 0$ and for μ in Theorem 1.1 that

$$\mu = \sum_{k=0}^{\infty}\frac{2p_k}{(k+1)(k+2)}. \tag{9}$$

Now, a similar derivation as for the previous case implies the integral representation for μ in (1). □

3. Domination number

We will show that we again can apply Corollary 1.15 in [19] (on normal limit laws for the number of fringe trees) to deduce normal limit laws for the domination number of random binary search trees and random recursive trees. Note that the domination number is not directly related to the independence number; in particular it is not an affine function of the independence number.

Proof of Theorem 1.4. Let T be a rooted tree with n vertices and let S be a minimum dominating set of T and let $D(T) := |S|$ be its size. There are potentially several minimum dominating sets, but we may order these sets by counting nodes in S that are at each depth from the root giving rise to a vector, where the ith component describes the number of nodes at depth i that belongs to S, and then put a lexicographical order on these vectors (that is, those with more nodes at a depth closer to the root are larger with respect to this order).

We now introduce the following descriptions of so-called *root-dependent* and *root-independent* trees. Let r be the root vertex of T. We say that T is *root-dependent* if $D(T \setminus r) = D(T) - 1$. If this is not true, i.e., $D(T \setminus r) > D(T) - 1$ we say that T is *root-independent*.

We now assume that S is a minimum dominating set of T with the property that it is the largest with respect to the lexicographical order described above. We will show that $v \in S$, if and only if, $T(v)$ is root-independent and v is included in a minimum dominating set of $T(v)$ (we call this the Property A) except for maybe the root vertex $r \in T$ (which is treated differently).

If we have a vertex $v \neq r$ that is not contained in S and $T(v)$ is a root-independent tree which has a minimum dominating set containing v, then all the vertices of S from $T(v)$ form a dominating set of $T(v) \setminus v$. Since $T(v)$ is root-independent we have that $D(T(v) \setminus v) \geq D(T(v))$, and thus we could replace the set $S \cap T(v)$ with a minimum dominating set of $T(v)$ that contains v without increasing the size of S. However, that would give a minimum dominating set with a larger lexicographical order than S, which is a contradiction to that S has the largest order.

On the other hand, if S contains a vertex $v \neq r$ that is not contained in a minimum dominating set of a root-independent tree $T(v)$, it could either be because the tree $T(v)$ is root-dependent or because every minimum dominating set of $T(v)$ excludes v. We now show that in both cases this gives a contradiction.

In the first case, that is, if $T(v)$ is root-dependent, we could remove v from S and replace it with its parent and then replace the elements of S coming from $T(v) \setminus v$ with a minimum dominating set of $T(v) \setminus v$. This does not increase the size of S and it is still dominating. However, this gives a minimum dominating set with a larger lexicographical order, hence the set S we had chosen is not the largest with respect to this order.

In the second case, that is, if $T(v)$ is root-independent, but no minimum dominating set of $T(v)$ contains v, then we could replace v with its parent, and use a minimum dominating set of $T(v)$ instead (since no minimum dominating set of $T(v)$ contained v, it must be inefficient to include v in S if we only wanted to dominate $T(v)$). Thus, we would again get a minimum dominating set with a larger lexicographical order, which is a contradiction.

Finally, we note that a similar discussion as above shows that the root r will be included in S if any of the subtrees T_i rooted at the children $i = 1, 2, \ldots, m$ of r is root-dependent, or if all of these subtrees have minimum dominating sets which all exclude their root i.

Now, the domination number of a rooted tree T can be covered as a function F in (3) as follows. We set f as the indicator function

$$f_{\mathrm{dom}}(T) := \begin{cases} 1, & \text{if the root of } T \text{ satisfies Property A,} \\ 0, & \text{otherwise.} \end{cases}$$

We write $F_{\mathrm{dom}}(T) = \sum_{v \in T} f_{\mathrm{dom}}(T(v))$ as in (3) and note that $|F_{\mathrm{dom}}(T) - D(T)| \leq 1$. This implies that the domination numbers $D_n := D(\mathcal{T}_n) = F_{\mathrm{dom}}(\mathcal{T}_n) + O(1)$ and $\widehat{D}_n := D(\Lambda_n) = F_{\mathrm{dom}}(\Lambda_n) + O(1)$ in distribution. We have $f_{\mathrm{dom}}(T) = O(1)$. Hence, Corollary 1.15 of Holmgren and Janson [19] implies the assertions of Theorem 1.4. Unfortunately, as mentioned before, we do not have a closed expression for ν and $\widehat{\nu}$ (but, as mentioned in Remark 1.5, the numerical value satisfies $\widehat{\nu} = 0.3745\ldots$), and neither for τ^2 and $\widehat{\tau}^2$; however, the positivity of these constants follows by computing the first few significant digits (see Remark 1.7(b)).

It is also clear that we can easily modify our proof to obtain similar results for the k-domination number as discussed in Remark 1.6. □

4. Clique cover number

Computing the clique cover number, see Remark 1.7(c), of a general graph is NP-hard [22], and it is also NP-hard to approximate it up to a factor $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [36]. However, it is well known that for triangle-free graphs, in particular trees, the clique cover number coincides with the independence number on trees, see [13]. Hence, the clique cover number of random binary search trees and random recursive trees is covered by Theorems 1.1 and 1.2. However, in this section we give a direct proof of Theorems 1.1 and 1.2 for the clique cover number to show that this parameter can also be captured by the fringe tree representation and Corollary 1.15 in [19].

Proof. For a tree T, consider $T(v)$ with root v and subtrees $T_1, \ldots, T_k$ with corresponding roots $v_1, \ldots, v_k$ that are the children of v. For such a tree $T(v)$, let $\mathcal{E}_v$ be the indicator event that there exists a subtree T_i and an optimal clique coloring of the vertices of T_i (that is, a coloring using a minimal number $C(T_i)$ of colors, so that every edge of the complement of T_i is such that its incident vertices get different colors) such that v_i is the only vertex with color 1 in T_i. We then set f as the indicator function

$$f_{\mathrm{cc}}(T(v)) := \begin{cases} 0, & \text{if } \mathcal{E}_v \text{ holds,} \\ 1, & \text{otherwise.} \end{cases}$$

We show now that the clique cover number of T is equal to the number of vertices that were assigned 1. Indeed, we will show that there exists an optimal clique coloring which uses that number of colors. This coloring will be constructed inductively over all layers bottom up. Moreover, we will simultaneously prove by induction that our coloring indeed is proper and optimal. The deepest layer contains the set of leaves. Every leaf is assigned 1 under f since there are no subtrees of the leaves. Clearly all leaves are adjacent in the complement, so the set of leaves forms a clique in the complement, and thus all leaves must have different colors. The base case is satisfied. Now, suppose inductively that for a layer ℓ with vertices $u_1, \ldots, u_{j_\ell}$, $\bigcup_{i=1}^{j_\ell} F_i$ is optimally colored (optimal in the sense of the clique cover number), where F_i is the forest corresponding to the union of subtrees (at level $\ell - 1$) pending from vertex u_i. Now, we color the u_i's as follows: assume that for F_i say t colors are used. Shift these t colors to the set $\{1, \ldots, t\}$ and then try all possible permutations of $\{1, \ldots, t\}$ to check whether there exists a permutation such that $\mathcal{E}_{u_i}$ holds. If there is a permutation such that $f(T(u_i))$ evaluates to 0, assign to u_i the same color before the shift of the colors that was used for the root that was assigned color 1 after shifting and permuting colors. Otherwise, assign to u_i a color which was not used yet. We have to show now that this coloring of $u_1, \ldots, u_{j_\ell}$ gives a proper and optimal coloring of $\bigcup_{i=1}^{j_\ell} T(u_i)$. First, we show that it is proper. Note that the $T(u_i)$'s are all colored properly by definition of the color of u_i and the induction hypothesis which implies that any two vertices k_1, k_2 from different trees in F_i have different colors. Moreover, again by induction hypothesis, any two vertices k_1, k_2 from different forests F_i have also different colors. Thus, it suffices to show that all u_i's are colored differently since the subgraph induced by these vertices form a clique in the complement. However, this is clear since the colors of the u_i's either come from F_i or are entirely new colors. Thus, the coloring is indeed proper. To show that the coloring is optimal, first note that the clique cover number is monotone under adding vertices: if two vertices need to be assigned different colors in a subtree (subforest), they still need to be assigned different colors after adding a new vertex. If a vertex u_j is assigned 0, then no new color is used for such a vertex, and this coloring remains optimal. If a vertex u_j is assigned 1, then note that u_j must obtain a color different from all other vertices except for possibly those that are roots of the pending subtree (since u_j is adjacent to all of them in the complement). If there were a coloring assigning u_j the same color as the root of a pending subtree (and no other vertex of the subtree), then after permuting the colors one could assign to such a root color 1, and to no other vertex in the subtrees of $T(u_j)$ has color 1, and hence u_j would be assigned 0, contradicting this possibility. Hence u_j must be assigned a new color, and the coloring remains optimal.

Hence, the clique cover number of T is given by $F_{\mathrm{cc}}(T) = \sum_{v \in T} f_{\mathrm{cc}}(T(v))$ as in (3). This implies that the clique cover numbers $C_n := C(\mathcal{T}_n) = F_{\mathrm{cc}}(\mathcal{T}_n)$ and $\widehat{C}_n := C(\Lambda_n) = F_{\mathrm{cc}}(\Lambda_n)$ in distribution. We have $f_{\mathrm{cc}}(T(v)) = O(1)$. Hence, Corollary 1.15 of Holmgren and Janson [19] implies the assertions of Theorems 1.1 and 1.2. □

Acknowledgments

The results of the present note were obtained during the Twelfth Annual Workshop on Probability and Combinatorics at McGill University's Bellairs Research Institute. The authors thank the participants, in particular Luc Devroye and Remco van der Hofstad, for helpful discussions on the present problem. The hospitality and support of the institute is also acknowledged. When later also discussing our results with Stephan Wagner he told us that he together with his Ph.D. student Kenneth Dadedzi independently has shown results similar of our Theorems 1.1–1.2, see [9]. We would finally also like to thank Johan Björklund for helpful discussions that improved the writing of the proof concerning the domination number.

References

[1] F. Araujo, J. Farinha, P. Domingues, G.C. Silaghi, D. Kondo, A maximum independent set approach for collusion detection in voting pools, J. Parallel Distrib. Comput. 71 (2011) 1356–1366.
[2] B. Baker, Approximation algorithms for NP-complete problems on planar graphs, J. ACM 41 (1) (1980) 153–180.
[3] C. Bazgan, B. Escoffier, V.Th. Paschos, Completeness in standard and differential approximation classes: Poly-(D)APX- and (D)PTAS-completeness, Theoret. Comput. Sci. 339 (2–3) (2005) 272–292.
[4] B. Bollobás, The independence ratio of regular graphs, Proc. Amer. Math. Soc. 83 (1981) 433–436.
[5] S. Butenko, P. Pardalos, I. Sergienko, V. Shylo, P. Stetsyuk, Finding maximum independent sets in graphs arising from coding theory, in: Proc. of 2002 Symposium on Applied Computing, ACM, 2002, pp. 542–546.
[6] H. Chen, J. Jost, Minimum vertex covers and the spectrum of the normalized Laplacian on trees, Linear Algebra Appl. 437 (4) (2012) 1089–1101.
[7] E. Cockayne, S. Goodman, S. Hedetniemi, A linear algorithm for the domination number of a tree, Inform. Process. Lett. 4 (2) (1975) 41–44.
[8] C. Cooper, M. Zito, An analysis of the size of the minimum dominating sets in random recursive trees, using the Cockayne-Goodman-Hedetniemi algorithm, Discrete Appl. Math. 157 (9) (2009) 2010–2014.
[9] K. Dadedzi, Analysis of Tree Spectra (Ph.D. dissertation), Stellenbosch University, 2018.
[10] L. Devroye, Limit laws for sums of functions of subtrees of random binary search trees, SIAM J. Comput. 32 (2003) 152–171.
[11] J.F. Fink, M.S. Jacobson, n-domination in graphs, in: Graph Theory with Applications to Algorithms and Computer Science, Wiley (1985), Kalamazoo, Mich., 1984, pp. 283–300.
[12] F.V. Fomin, F. Grandoni, D. Kratsch, A measure & conquer approach for the analysis of exact algorithms, J. ACM 56 (5) (2009) 25, 1–32.
[13] W. Goddard, S.M. Hedetniemi, S.T. Hedetniemi, Eternal security in graphs, J. Combin. Math. Combin. Comput. 52 (2005) 160–180.
[14] M. Grohe, Local tree-width, excluded minors, and approximation algorithms, Combinatorica 23 (4) (2003) 613–632.
[15] M. Grötschel, L. Lovász, A. Schrijver, Geometric Algorithms and Combinatorial Optimization, in: Algorithms and Combinatorics, vol. 2, Springer, 1988.
[16] T.W. Haynes, S.T. Hedetniemi, P.J. Slater, Fundamentals of Domination in Graphs, in: Monographs and Textbooks in Pure and Applied Mathematics, vol. 208, Marcel Dekker, New York, NY, USA, 1998.

[17] T.W. Haynes, S.T. Hedetniemi, P.J. Slater (Eds.), Domination in Graphs, in: Monographs and Textbooks in Pure and Applied Mathematics, vol. 209, Marcel Dekker, New York, NY, USA, 1998.
[18] S.T. Hedetniemi, R.C. Laskar, Bibliography on domination in graphs and some basic definitions of domination parameters, Discrete Math. 86 (1–3) (1990) 257–277.
[19] C. Holmgren, S. Janson, Limit laws for functions of fringe trees for binary search trees and random recursive trees, Electron. J. Probab. 20 (4) (2015) 1–51.
[20] H.-K. Hwang, R. Neininger, Phase change of limit laws in the quicksort recurrence under varying toll functions, SIAM J. Comput. 31 (2002) 1687–1722.
[21] C. Joo, X. Lin, J. Ryu, N.B. Shroff, Distributed greedy approximation to maximum weight independent set for scheduling with fading channels, IEEE/ACM Trans. Netw. 24 (2016) 1476–1488.
[22] R. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Proc. of a Symposium on the Complexity of Computer Computations, Plenum Press, 1972, pp. 85–103.
[23] D.E. Knuth, The Art of Computer Programming, second ed., in: Sorting and Searching, vol. 3, Addison-Wesley, 1998.
[24] D. Lokshtanov, M. Vatshelle, Y. Villanger, Independent sets in P5-free graphs in polynomial time, in: SODA (Symposium on Discrete Algorithms), 2014, pp. 570–581.
[25] A. Meir, J.W. Moon, The expected node-independence number of various types of trees, in: Recent Advances in Graph Theory, Proc. Second Czechoslovak Sympos., Prague, 1974, Academia, Prague, 1975, pp. 351–363.
[26] T. Milenković, V. Memišević, A. Bonato, N. Pržulj, Dominating biological networks, PLoS One 6 (8), e23016.
[27] X. Mingyu, H. Nagamochi, Confining sets and avoiding bottleneck cases: A simple maximum independent set algorithm in degree-3 graphs, Theoret. Comput. Sci. 469 (2013) 92–104.
[28] X. Mingyu, H. Nagamochi, Exact algorithms for maximum independent set, Inform. Comput. 255 (2017) 126–146.
[29] G.J. Minty, On maximal independent sets of vertices in claw-free graphs, J. Combin. Theory Ser. B 28 (3) (1980) 284–304.
[30] R. Raz, S. Safra, A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP, in: 29th Symposium on Theory of Computing, STOC, ACM, 1997, pp. 475–484.
[31] J.M. Robson, Algorithms for maximum independent sets, J. Algorithms 7 (1986) 425–440.
[32] R.T. Smythe, H.M. Mahmoud, A survey of recursive trees, Teor. Ĭmovīr. Mat. Stat. 51 (1994) 1–29.
[33] I. Stojmenović, M. Seddigh, J. Zunić, Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks, IEEE Trans. Parallel Distrib. Systems 13 (2002) 14–25.
[34] K. Takamizawa, T. Nishizeki, N. Saito, Linear-time computability of combinatorial problems on series-parallel graphs, J. ACM 29 (3) (1982) 623–641.
[35] J.M.M. van Rooij, J. Nederlof, T.C. van Dijk, Inclusion/exclusion meets measure and conquer: Exact algorithms for counting dominating sets, in: Proc. 17th Annual European Symposium on Algorithms, ESA, in: Lecture Notes in Computer Science, vol. 5757, Springer, 2009, pp. 554–565.
[36] D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, Theory Comput. 3 (2007) 103–128.