Greedy maximal independent sets via local limits

Michael Krivelevich∗ Tamás Mészáros† Peleg Michaeli‡ Clara Shikhelman§

July 12, 2021

Abstract

The random greedy algorithm for finding a maximal independent set in a graph has been studied extensively in various settings in combinatorics, probability, computer science, and even chemistry. The algorithm builds a maximal independent set by inspecting the graph’s vertices one at a time according to a random order, adding the current vertex to the independent set if it is not connected to any previously added vertex by an edge.

In this paper, we present a natural and general framework for calculating the asymptotics of the proportion of the yielded independent set for sequences of (possibly random) graphs, involving a notion of local convergence. We use this framework both to give short and simple proofs for results on previously studied families of graphs, such as paths and binomial random graphs, and to study new ones, such as random trees and random planar graphs.

We conclude our work by analysing the random greedy algorithm more closely when the base graph is a tree. We show that in expectation, the cardinality of a random greedy independent set in the path is no larger than that in any other tree of the same order.

1 Introduction

Algorithmic problems related to finding or approximating the independence number of a graph, or to producing large independent sets, have long been in the focus of the computer science community. Computing the size of a maximum independent set is known to be NP-complete [33], and the groundbreaking work [18] on the difficulty of approximating it even made its way to The New York Times. A natural way to try to efficiently produce a large independent set in an input graph $G$ is to output a maximal independent set (MIS) in $G$, where a vertex subset $I \subseteq V(G)$ is a MIS in $G$ if $I$ is maximal by inclusion. While in principle a poorly chosen MIS can be very small (like, say, the star centre in a star), one might hope that quite a few of the maximal independent sets will have a size comparable in some quantitative sense to the independence number of $G$.

∗School of Mathematical Sciences, Tel Aviv University, Tel Aviv 6997801, Israel. E-mail: krivelev@tauex.tau.ac.il. Research supported in part by USA-Israel BSF grant 2018267, and by ISF grant 1261/17.
†Fachbereich Mathematik und Informatik, Kombinatorik und Graphentheorie, Freie Universität Berlin, Arnimallee 3, 14195 Berlin, Germany. E-mail: meszaros.tamas@fu-berlin.de. Research supported by the Berlin Mathematics Research Center MATH+.
‡School of Mathematical Sciences, Tel Aviv University, Tel Aviv 6997801, Israel. E-mail: peleg.michaeli@math.tau.ac.il. Research supported by ERC starting grant 676970 RANDGEOM and by ISF grants 1207/15, 1261/17 and 1294/19.
§Chaincode Labs, 450 Lexington Avenue, New York, NY 10017, USA. E-mail: clara.shikhelman@gmail.com.
This paper studies the random greedy algorithm for producing a maximal independent set, which is defined as follows. Consider an input graph $G$ on $n$ vertices. The algorithm first orders the vertices of $G$ uniformly at random and then builds an independent set $I(G)$ by considering each of the vertices $v$ one by one in order, adding $v$ to $I(G)$ if the resulting set does not span any edge. Observe that the set $I(G)$ is, in fact, the set of vertices coloured in the first colour in a random greedy proper colouring of $G$. A basic quantity to study, which turns out to have numerous applications, is the proportion of the yielded independent set (which we call the greedy independence ratio). In particular, it is of interest to study the asymptotic behaviour of this quantity for natural graph sequences.

Due to its simplicity, this greedy algorithm has been studied extensively by various authors in different fields, ranging from combinatorics [54], probability [48] and computer science [20] to chemistry [22]. As early as 1931, this model was studied by chemists under the name random sequential adsorption (RSA), focusing primarily on $d$-dimensional grids. The 1-dimensional case was solved by Flory [22] (see also [44]), who showed that the expected greedy independence ratio tends to $\zeta_2 = (1 - e^{-2})/2$ as the path length tends to infinity.

A continuous analogue, in which “cars” of unit length “park” at random free locations on the interval $[0, X]$, was introduced (and solved) by Rényi [49], under the name car-parking process. The limiting density, as $X$ tends to infinity, is therefore called Rényi’s parking constant, and $\zeta_2$ may be considered as its discrete counterpart (see, e.g., [19]). Following this terminology, the final state of the car-parking process is often called the jamming limit of the graph, and the density of this state is called the jamming constant. For dimension 2, Palásti [45] conjectured, in the continuous case (where unit square “cars” park in a larger square), that the limiting density is Rényi’s parking constant squared. This conjecture may be carried over to the discrete case, but to the best of our knowledge, in both cases, it remains open. For further details, see [19] (see also [17] for an extensive survey on RSA models).

In combinatorics, the greedy algorithm for finding a maximal independent set was analysed in order to give a lower bound on the (usually asymptotic) typical independence number of (random) graphs. The asymptotic greedy independence ratio of binomial random graphs with linear edge density was studied by McDiarmid [39] (but see also [9,27]). The asymptotic greedy independence ratio of random regular graphs was studied by Wormald [54], who used the so-called differential equation method (see [55] for a comprehensive survey; see also [53] for a short proof of Wormald’s result). His result was further extended in [37] for any sequence of regular graphs with growing girth (see also [30,31] for similar extensions for more sophisticated algorithms). Recently, the case of uniform random graphs with given degree sequences was studied (independently) in [5] and [11].

In a more general setting, where we run the random greedy algorithm on a hypergraph, the model recovers, in particular, the triangle-free process (or, more generally, the $H$-free process). In this process, which was first introduced in [16], we begin with the empty graph, and at each step, add a random edge as long as it does not create a copy of a triangle (or a copy of $H$). To recover this process, we take the hypergraph whose vertices are the edges of the complete graph and whose hyperedges are the triples of edges that span a triangle (or $k$-sets of edges).
that form a copy of $H$, if $H$ has $k$ edges). Bohman’s key result [7] is that for this hypergraph, $|I|$ is with high probability (whp)\(^3\) $\Theta(n^{3/2}\sqrt{\ln n})$, where $n$ is the number of vertices. Bohman and Keevash [8] and Fiz Pontiveros, Griffiths and Morris [21] later found the exact asymptotics. Similar results were obtained for the complete graph on four vertices by Warnke [52] and for cycles independently by Picollelli [47] and by Warnke [51]. For a discussion about the general setting, see [4].

Consider the following alternative but equivalent definition of the model. Assign an independent uniform label from $[0, 1]$ to each vertex of the graph, and consider it as the arrival time of a particle at that vertex. All vertices are initially vacant, and a vertex becomes occupied at the time denoted by its label if and only if all of its neighbours are still vacant at that time. Clearly, we do not need to worry that two particles will arrive at the same time. The set of occupied vertices at time 1 is exactly the greedy MIS. One advantage of this formulation of the model is that under mild assumptions, it can be defined on an infinite graph. We may think of the resulting MIS as a factor of iid (fiid)\(^4\), meaning, informally, that there exists a local rule which is unaware of the “identity” of a given vertex, that determines whether that vertex is occupied. It was conjectured (formally by Hatami, Lovász and Szegedy [28]) that, using a proper rule, fiid can produce an asymptotically maximum independent set in random regular graphs. However, this was disproved by Gamarnik and Sudan [25]. In fact, they showed that this kind of local algorithms has a uniformly limited power for a sufficiently large degree, and later Rahman and Virág [48] showed that the density of fiid independent sets in regular trees and Poisson Galton–Watson trees, with a large average degree, is asymptotically at most half-optimal, concluding (after projecting to random regular graphs or to binomial random graphs) that local algorithms cannot achieve better. In particular, this implies that the random greedy algorithm is, asymptotically, at least as good as any other local algorithm.

However, on other families of graphs, local algorithms may do better than that. A trivial example is the set of stars, where the greedy algorithm typically performs perfectly. A less trivial example is that of uniform random trees. The expected independence ratio of a uniform random tree is the unique solution of the equation $x = e^{-x}$ (see [40]), which is approximately $0.5671...$, while the greedy algorithm yields an independent set of expected density $1/2$ as we will see in Section 6.3.

Finally, we note that the following parallel/distributed algorithm gives a further way to look at the maximal independent set generated by the greedy algorithm. After (randomly) ordering the vertices, we colour “red” all the sinks, that is, all the vertices which appear before their neighbours in the order, and then remove them and their neighbours from the graph and continue. Formulated this way, the algorithm is straightforward to implement and requires only local communication between the nodes. Also, conditioning on the initial random ordering, it is deterministic, a property that appears to be important (see, e.g., [6]). A central question of interest is the number of rounds it takes the algorithm to terminate. In [20] it was shown that it terminates in $O(\log n)$ steps whp on any $n$-vertex graph, and that this is tight. Thus, even though these algorithms may be suboptimal, they are strikingly simple and surprisingly efficient.

\(^3\)That is, with probability tending to 1 as $n$ tends to infinity.

\(^4\)The letters iid abbreviate “independent and identically distributed”.
1.1 Our contribution

This work aims to introduce a simple and fairly general framework for calculating the asymptotics of the greedy independence ratio for a wide variety of (random) graph sequences. The general approach is to study a suitable limiting object, typically a random rooted infinite graph, which captures the local view of a typical vertex, and to calculate the probability that its root appears in a random independent set in this graph, which is created according to some natural “local” rule, to be described later. We show that this probability approximates the expected greedy independence ratio and, in many cases of interest, gives tools to calculate that probability precisely. This change of view from a sequence of graphs with varying underlying probability spaces to an infinite object with a fixed underlying probability space is known in the literature as the objective method [2].

Let us formulate this more precisely. For a (random) finite graph $G$ let $I(G)$ be the random greedy maximal independent set of $G$, let $\iota(G) := |I(G)|/|V(G)|$ be its density, and let $\bar{\iota}(G)$ be its expected density (taken over the distribution of $G$ and over the random greedy maximal independent set). Suppose $(U, \rho)$ is a random rooted locally finite graph (that is, $(U, \rho)$ is a distribution on rooted locally finite graphs). A random labelling $\sigma = (\sigma_v)_{v \in V(U)}$ of $U$ is a process consisting of iid random variables $\sigma_v$, each distributed uniformly in $[0,1]$. The past of a vertex $v$, denoted $P_v$, is the set of vertices in $U$ reachable from $v$ by a monotone decreasing path (with respect to $\sigma$). We say that $(U, \rho)$ has nonexplosive growth if the past of $\rho$ is almost surely finite. For such $(U, \rho)$ we may define

$$\iota(U, \rho) = \mathbb{P}[\rho \in I(U[P_\rho])].$$

We say that a graph sequence $G_n$ converges locally to $(U, \rho)$, and denote it by $G_n \xrightarrow{\mathrm{loc}} (U, \rho)$, if for every $r \geq 0$, the ball of radius $r$ around a uniformly chosen point from $G_n$ converges in distribution to the ball of radius $r$ around $\rho$ in $U$. To make this notion precise, we need to endow the space of rooted locally finite connected graphs with a topology. This will be done rigorously in Section 2. The following key theorem gives motivation for the definitions above.

**Theorem 1.1.** If $G_n \xrightarrow{\mathrm{loc}} (U, \rho)$ and $(U, \rho)$ has nonexplosive growth then $\bar{\iota}(G_n) \rightarrow \iota(U, \rho)$.

We remark that $\iota(U, \rho)$ is almost surely positive (it is at least $\mathbb{E}[(d(\rho) + 1)^{-1}]$), implying that for locally convergent graph sequences, the expected size of the random greedy maximal independent set is linear.

With some mild growth assumptions on the graph sequence, we can also obtain asymptotic concentration of the greedy independence ratio around its mean. For a graph $G$ let $N_G(r)$ be the random variable counting the number of paths of length at most $r$ from a uniformly chosen random vertex of $G$. For two real numbers $x, y$ denote by $x \land y$ their minimum. For a graph sequence $G_n$, let

$$\mu^*(r) = \lim_{M \to \infty} \limsup_{n \to \infty} \mathbb{E}[N_{G_n}(r) \land M].$$

We say that $G_n$ has subfactorial path growth (sfpg) if $\mu^*(r) \ll_r r!$. Note that every graph sequence with uniformly bounded degrees has sfpg, but there are graph sequences with unbounded degrees, and even with unbounded average degree, which still have sfpg. For most cases, and for all of the applications presented in this paper, requiring that the somewhat simpler expression $\limsup_{n \to \infty} \mathbb{E}[N_{G_n}(r)]$ is subfactorial would have sufficed; however, requiring

\[\text{By } g_1(r) \ll_r g_2(r) \text{ we mean that } \lim_{r \to \infty} g_1(r)/g_2(r) = 0.\]
that $\mu^*(r)$ is subfactorial is less strict, and is more natural for the following reason: if the graph sequence converges locally, then $\mu^*(r)$ is the expected number of paths of length at most $r$ in the limit. For two functions $f_1(n), f_2(n)$ write $f_1(n) \sim f_2(n)$ if $f_1(n) = (1 + o(1))f_2(n)$. We are now ready to state our concentration result.

**Theorem 1.2.** If $G_n$ has sfpg and $G_n \loc \to (U, \rho)$ then $\iota(G_n) \sim \iota(U, \rho)$ with high probability.

**Remark.** Gamarnik and Goldberg [24] have established concentration of $\iota(G_n)$ around its mean, under the assumption that the degrees of $G_n$ are uniformly bounded. Here we relax that assumption.

**Remark.** A sequence of graphs with sfpg does not necessarily have a local limit, but it does have a locally convergent subsequence. Any limit of such a sequence will have nonexplosive growth.

When the limiting object is supported on rooted trees, we call the (random) graph sequence **locally tree-like.** Our next result is a general differential-equations based tool for analysing the asymptotics of the greedy independence ratio of locally tree-like (random) sfpg graph sequences, with the restriction that their limit may be emulated by a simple branching process with at most countably many types. Roughly speaking, a **multitype branching process** is a rooted tree, in which each node is assigned a **type** and the number and types of each node’s “children” follow a law that depends solely on the node’s type, and is independent for distinct nodes. Such a branching process is called **simple** if each such law is a product measure. Formal definitions will be given in Section 4. The following theorem reduces the problem of calculating $\iota(U, \rho)$ in these cases to the problem of solving a (possibly infinite) system of ODEs.

In the following, given a (countable) set of types $T$, for every two types $k, j \in T$ we denote by $\mu_k \to j$ the distribution of the number of nodes of type $j$ to a parent of type $k$. Roughly speaking, a multitype branching process is a rooted tree, in which each node is assigned a type and the number and types of each node’s “children” follow a law that depends solely on the node’s type, and is independent for distinct nodes. Such a branching process is called simple if each such law is a product measure. Formal definitions will be given in Section 4. The following theorem reduces the problem of calculating $\iota(U, \rho)$ in these cases to the problem of solving a (possibly infinite) system of ODEs.

**Theorem 1.3.** Let $(U, \rho)$ be a simple multitype branching process with finite or countable type set $T$, root distribution $\hat{\mu}$ and offspring distributions $\mu_k \to j$. For every $x \in [0, 1]$ and $k, j \in T$ let $\mu_{x}^{k \to j} = \text{Bin}(\mu_{x}^{k \to j}, x)$ denote the distribution of the number of children of type $j$ of a node of type $k$ with random label at most $x$. Then,

$$\iota(U, \rho) = \sum_{k \in T} y_k(1) \hat{\mu}(k),$$

where $\{y_k\}_{k \in T}$ is a solution to the following system of ODEs:

$$y_k(x) = \sum_{\ell \in \mathbb{N}^T} \prod_{j \in T} \mu_{x}^{k \to j}(\ell_j) \left(1 - \frac{y_j(x)}{x}\right)^{\ell_j}, \quad y_k(0) = 0. \quad (*)$$

We call $(*)$ the **fundamental system of ODEs** of the branching process $(U, \rho)$. While this system of ODEs may seem complicated, in many important cases it reduces to a fairly simple system, as we will demonstrate in Section 6. In particular, the proof of Theorem 1.3 implies that a solution to $(*)$ exists, and in the presented applications, it will be unique. In the cases where $(U, \rho)$ is either a single type branching process or a random tree with iid degrees, we provide an easy probability generating function tool that may be used to “skip” solving $(*)$. This is described in Section 5. We mention that a somewhat related, but apparently less applicable statement, providing differential equations for the occupancy probability of a given vertex in bounded degree graphs, appears in [46].
Remark. The proof of Theorem 1.3 actually yields a stronger result. Replacing \( y_k(1) \) with \( y_k(x) \) in the RHS of (1), the obtained quantity is the probability that the root is occupied “at time \( x \)”, namely, when vertices whose label is above \( x \) are ignored.

Remark. As mentioned in the introduction, one may run the random greedy algorithm for producing a maximal independent set on a hypergraph. Here, an independent set is a set of vertices that does not span a hyperedge. The formal definitions for local convergence (see Section 2) and \( \text{sfpg} \) easily generalise to the hypergraph setting, thus Theorems 1.1 and 1.2 are also valid in this setting. In fact, it is not hard to generalise the notion of simple multitype branching processes to represent local limits of hypergraphs. Hence (an analogue of) Theorem 1.3 can also be applied in this setting. In Section 7, we discuss how to calculate the asymptotic size of the random greedy maximal independent set on locally tree-like hypergraphs.

We conclude our work with a theorem, according to which on the set of all trees of a given order, the expected size of the greedy MIS achieves its minimum on the path.

**Theorem 1.4.** Let \( n \geq 1 \), let \( T \) be a tree on \( n \) vertices and let \( P_n \) be the path on \( n \) vertices. Then \( \bar{\iota}(P_n) \leq \bar{\iota}(T) \).

This theorem gives us an exact (non-asymptotic) explicit lower bound for the expected greedy independence ratio of trees (an asymptotic upper bound of 1 is trivial, as can be seen by considering the sequence of stars). The methods used to prove it are very different from those used in the rest of this paper and are more combinatorial. In particular, we use a transformation on trees, initially introduced by Csikvári in [13], which gives rise to a graded poset of all trees of a given order, in which the path is the unique minimum (say). While we cannot show that this transformation can only increase the expected greedy independence ratio, we show it can only increase some other quantitative property of trees, which allows us to argue that paths indeed achieve the minimum expected greedy independence ratio.

**1.2 Organisation of the paper**

We start with formal definitions and proofs of the main results. We introduce the metric that is used to define the notion of local convergence in Section 2, where we also prove Theorem 1.1. In Section 3, we prove Theorem 1.2 by essentially proving a decay of correlation between vertices in terms of their distance and showing that typical pairs of vertices are distant. In fact, the results of Section 3 imply that even without local convergence, under mild growth assumptions, the variance of the greedy independence ratio is decaying.

In Section 4, we focus our attention on locally tree-like graph sequences, define (simple, multitype) branching processes, and prove Theorem 1.3. We enhance this in Section 5 by introducing a probability generating functions based “trick”, which allows, in some cases, a significant simplification.

We continue by presenting a non-exhaustive list of important applications in Section 6, where we prove some new results and reprove some known ones, using the machinery of Theorems 1.2 and 1.3. In a few cases, we are assisted by the claims from Section 5. In particular, we calculate the asymptotics of the greedy independence ratio for paths and cycles (reproving results from [22,44]), binomial random graphs (reproving a result from [39]), uniform spanning trees and random functional digraphs (new results), sparse random planar graphs (new results) and random regular graphs or regular graphs with high girth (reproving results from [37,54]). In Section 7, we demonstrate how the presented tools work, almost as-is, for locally tree-like hypergraphs, by reproving results from a recent paper by Nie and Verstraëte [43].
In Section 8 we focus further on tree sequences, where we prove Theorem 1.4. To this end, we pinpoint several interesting properties of the expected greedy independence ratio of the path.

2 Local limits

In order to study asymptotics, it is often useful to construct a suitable limiting object first. Local limits were introduced by Benjamini and Schramm [3] and studied further by Aldous and Steele [2] (A very similar approach has already been introduced by Aldous in [1]). Local limits, when they exist, encapsulate the asymptotic data of local behaviour of the convergent graph sequence, and in particular, that of the performance of the greedy algorithm.

We start with basic definitions, which we define for graphs but that can be extended to hypergraphs in an obvious way. Consider the space $G_{\bullet}$ of rooted locally finite connected graphs viewed up to root preserving graph isomorphisms. We provide $G_{\bullet}$ with the metric $d_{\text{loc}}((G_1, \rho_1), (G_2, \rho_2)) = 2^{-R}$, where $R$ is the largest integer for which $B_{G_1}(\rho_1, R) \simeq B_{G_2}(\rho_2, R)$. Here we understand $B_{G}(\rho, R)$ as the rooted subgraph of $(G, \rho)$ spanned by the vertices of distance at most $R$ from $\rho$, and $\simeq$ as rooted-isomorphic. It is an easy fact that $(G_{\bullet}, d_{\text{loc}})$ is a separable complete metric space, hence it is a Polish space. $(G_{\bullet}, d_{\text{loc}})$, while being bounded, is not compact (the sequence of rooted stars $S_n$ does not have a convergent subsequence).

Recall that a sequence of random elements $\{X_n\}_{n=1}^{\infty}$ converges in distribution to a random element $X$, if for every bounded continuous function $f$ we have that $E[f(X_n)] \to E[f(X)]$. Let $G_n$ be a sequence of (random) finite graphs. We say that $G_n$ converges locally to a (random) element $(U, \rho)$ of $G_{\bullet}$ if for every $r \geq 0$, the sequence $B_{G_n}(\rho_n, r)$ converges in distribution to $B_U(\rho, r)$, where $\rho_n$ is a uniformly chosen vertex of $G_n$. Since the inherited topology on all rooted balls in $G_{\bullet}$ with radius $r$ is discrete, this implies convergence in total variation distance.

We are now ready to prove Theorem 1.1.

Proof of Theorem 1.1. Fix $\varepsilon > 0$. For a given labelling $\pi$ of $U$, let $\ell_{\pi}$ be the length of the longest decreasing sequence (w.r.t. $\pi$) starting from $\rho$. Since $(U, \rho)$ has nonexplosive growth, there exists $r_\varepsilon$ for which for every $r \geq r_\varepsilon$, $P[\ell_{\pi} \geq r] < \varepsilon$. For $r \geq 0$, let $G_n^r = B_{G_n}(\rho_n, r)$ and $U^r = B_U(\rho, r)$. We couple $G_n^r$ and a random permutation $\pi$ on its vertices with $U^r$ and a random labelling $\pi$ as follows. First, since $G_n^r$ converges in distribution (and hence in total variation distance) to $U^r$, there exists $n_r$ such that for all $n \geq n_r$ we have that $P[G_n^r \not\simeq U^r] \leq \varepsilon$. If this event occurs, we say that the coupling has failed. Otherwise, for some isomorphism $\varphi : G_n^r \to U^r$, we let $\pi$ be the permutation on the vertices of $G_n^r$ which agrees with the ordering of the labels on the vertices of the isomorphic image (that is, $\pi_u < \pi_v \iff \varphi(u) < \varphi(v)$). Note that under this coupling, if it succeeds, $\rho_n \in I(G_n^r) \iff \rho \in I(U^r)$. However, on the event $\ell_{\pi} \leq r_\varepsilon$, $\rho_n \in I(G_n^r) \iff \rho_n \notin I(G_n)$ and $\rho \in I(U^r) \iff \rho \notin I(U[I(P_\rho)])$. Observing that $E[|\tilde{I}(G_n) - \tilde{I}(U, \rho)| < 2\varepsilon].$

3 Concentration via exploration–decision algorithms

With some mild growth assumptions on the graph sequence, without assuming local convergence, we obtain asymptotic concentration of the greedy independence ratio around its mean. Under these assumptions we show that the dependence between the inclusion of distinct nodes in the maximal independent set decays as a functions of their distance, a phenomenon which
is sometimes called correlation decay or long-range independence. To prove that the model exhibits this phenomenon, we show that with high probability there are no “long” monotone paths emerging from a typical vertex, which is the contents of the next claim. We then observe that two independent random vertices are typically distant, and use a general lemma about exploration algorithms to prove decay of correlation. We remark that similar locality arguments appear in [42].

**Claim 3.1.** Suppose that \( G_n \) has sfpg. Let \( \pi \) be a uniform random permutation of the vertices of \( G_n \), and let \( u \) be a uniformly chosen vertex from \( G_n \). Then, for every \( \varepsilon > 0 \), there exists \( r > 0 \) such that for every large enough \( n \), the probability that there exists a monotone decreasing path of length \( r \) w.r.t. \( \pi \), emerging from \( u \), is at most \( \varepsilon \).

**Proof.** Let \( \varepsilon \geq 0 \). Since \( \mu^*(r) \ll r! \) for every large enough \( r \) we have \( \mu^*(r) \leq \varepsilon r! \). We couple \( \mathcal{N}_{G_n}(r) \) and \( u \) such that the former counts the number of paths of length at most \( r \) emerging from the latter. Denote by \( A^*_r \) the event that there exists a monotone decreasing path in \( G_n \) (w.r.t. \( \pi \)) emerging from \( u \) of length \( r \). Note that the probability that a given path of length \( r \) is monotone decreasing w.r.t. \( \pi \) is \( 1/r! \). Since \( \mu^*(r) \) is finite, there exists \( M \geq 0 \) such that \( \mathbb{P}[\mathcal{N}_{G_n}(r) > M] < \varepsilon \) for every large enough \( n \). In addition, for large enough \( n \) we have \( \mathbb{E}[\mathcal{N}_{G_n}(r) \wedge M] \leq 2\mu^*(r) \). Hence, for large enough \( n \),

\[
\mathbb{P}[A^*_r] \leq \sum_{m=0}^{M} \mathbb{P}[A^*_n | \mathcal{N}_{G_n}(r) = m] \cdot \mathbb{P}[\mathcal{N}_{G_n}(r) = m] + \mathbb{P}[\mathcal{N}_{G_n}(r) > M]
\leq \frac{1}{r!} \cdot \mathbb{E}[\mathcal{N}_{G_n}(r) \wedge M] + \varepsilon \leq 3\varepsilon. \quad \Box
\]

**Claim 3.2.** Suppose that \( G_n \) has sfpg. Let \( u, v \) be two independently and uniformly chosen vertices from \( G_n \). Then, for every \( \varepsilon, r \geq 0 \), we have that for every large enough \( n \), \( \mathbb{P}[\text{dist}_{G_n}(u, v) \leq r] \leq \varepsilon \).

**Proof of Claim 3.2.** Let \( \varepsilon, r \geq 0 \). We couple \( \mathcal{N}_{G_n}(r) \) and \( u \) such that the former counts the number of paths of length at most \( r \) emerging from the latter. Note that under this coupling, \( |B_{G_n}(u, r)| \leq \mathcal{N}_{G_n}(r) \). Since \( \mu^*(r) \) is finite, there exists \( M \geq 0 \) such that \( \mathbb{P}[\mathcal{N}_{G_n}(r) > M] < \varepsilon \) for every large enough \( n \). Hence, for large enough \( n \),

\[
\mathbb{P}[\text{dist}_{G_n}(u, v) \leq r] = \mathbb{P}[v \in B_{G_n}(u, r)]
\leq \sum_{m=0}^{M} \mathbb{P}[v \in B_{G_n}(u, r) | \mathcal{N}_{G_n}(r) = m] \cdot \mathbb{P}[\mathcal{N}_{G_n}(r) = m] + \mathbb{P}[\mathcal{N}_{G_n}(r) > M]
\leq \frac{1}{n} \cdot \mathbb{E}[\mathcal{N}_{G_n}(r) \wedge M] + \varepsilon \leq \frac{M}{n} + \varepsilon \leq 2\varepsilon. \quad \Box
\]

**Remark.** We only used the fact that \( \mathcal{N}_{G_n}(r) \) are uniformly integrable for every \( r \geq 0 \).

Let \( G = (V, E) \) be a graph. An **exploration-decision rule** for \( G \) is a (deterministic) function \( Q \), whose input is a pair \((S, g)\), where \( S \) is a non-empty sequence of distinct vertices of \( V \), and \( g : S \to [0, 1] \), and whose output is either a vertex \( v \in V \setminus S \) or a “decision” \( T \) or \( F \). An **exploration-decision algorithm** for \( G \), with rule \( Q \), is a (deterministic) algorithm \( A \), whose input is an initial vertex \( v \in V \) and a function \( f : V \to [0, 1] \), which outputs \( T \) or \( F \), and operates as follows. Set \( u_1 = v \). Suppose \( A \) has already set \( u_1, \ldots, u_i \). Let \( x = Q((u_1, \ldots, u_i), f |_{\{u_1, \ldots, u_i\}}) \). If \( x \in V \), set \( u_{i+1} = x \) and continue. Otherwise stop and
return $x$. We call the set $u_1, \ldots, u_i$ at this stage the range of the algorithm’s run. We denote the output of the algorithm by $A(v, f)$ and its range by $\text{rng}_A(v, f)$. The radius of the algorithm’s run, denoted $\text{rad}_A(v, f)$, is the maximum distance between $v$ and an element of its range.

**Lemma 3.3.** Let $\epsilon > 0$. Let $G = (V, E)$ be a graph, let $\sigma$ be a random labelling of its vertices, let $A$ be an exploration–decision algorithm for $G$ and let $r \geq 1$. Let $u, v$ be sampled independently from some distribution over $V$. Suppose that w.p. at least $1 - \epsilon$ both $\text{dist}_G(u, v) \geq 3r$, and $\text{rad}_A(u, \sigma), \text{rad}_A(v, \sigma) \leq r$. Then $|\text{cov}[A(u, \sigma), A(v, \sigma)]| = o_\epsilon(1)$.

**Proof.** Let $Q$ be the rule of the algorithm $A$. The $r$-truncated version of $Q$, denoted $Q^r$, is defined as follows. To determine $Q^r((u_1, \ldots, u_i), g)$, $Q^r$ checks the value $x = Q((u_1, \ldots, u_i), g)$. If $x \in \{T, F\}$ or $\text{dist}_G(u_1, x) \leq r$, $Q$ returns $x$. Otherwise it returns $F$. The $r$-truncated version of the algorithm $A$, denoted $A^r$, is the exploration–decision algorithm with rule $Q^r$. Note that for every $v$ and $f$, $\text{rad}_{A^r}(v, f) \leq r$.

For a vertex $w \in \{u, v\}$, let $X_w$ be the event “$A(w, \sigma) = T$”, let $Y_w$ be the event “$A^r(w, \sigma) = T$”, and let $r_w = \text{rad}_A(w, \sigma)$. Note that $P[X_w \land r_w \leq r] = P[Y_w \land r_w \leq r] = P[Y_w]$, thus $P[X_w] = P[Y_w] + o_\epsilon(1)$. Since for $x, y$ satisfying $\text{dist}_G(x, y) \geq 3r$ we have that $Y_x, Y_y$ are independent, it follows that $P[Y_u \land Y_v] = P[Y_u]P[Y_v] + o_\epsilon(1)$.

We now apply the lemma in our setting.

**Claim 3.4.** Suppose that $G_n$ has sfpg. Let $u, v$ be two independently and uniformly chosen vertices from $G_n$. Denote by $R_u, R_v$ the events that $u \in I(G_n), v \in I(G_n)$, respectively. Then $|\text{cov}[R_u, R_v]| = o(1)$.

**Proof.** Let $\epsilon > 0$. We describe an exploration–decision algorithm $A$ by defining its rule. Given a vertex sequence $S = (u_1, \ldots, u_i)$ and labels $g : S \rightarrow [0, 1]$, the rule checks for monotone decreasing sequences emerging from $u_1$, in $S$, with respect to $g$. Denote by $E$ the set of ends of these sequences. If there are vertices in $V \setminus S$ with neighbours in $E$, return an arbitrary vertex among these. Otherwise, perform the Greedy MIS algorithm on the past of $u_1$ inside $S$, and return $T$ if $u_1$ ends up in the MIS, or $F$ otherwise. We observe that if $\sigma$ is a random labelling of $G_n$ then for $w \in \{u, v\}$ the event $A(w, \sigma) = T$ is in fact the event $R_w$. We also note that if the longest monotone decreasing sequence, w.r.t. $\sigma$, emerging from $w$ is of length $r - 1$, then $\text{rad}_A(w, \sigma) \leq r$.

By Claim 3.1 there exists $r > 0$ such that for every large enough $n$ the probability that there exists a monotone decreasing path of length $r - 1$ from either $u$ or $v$ is at most $\epsilon$. By Claim 3.2, for large enough $n$, the probability that the distance between $u$ and $v$ is at most $3r$ is at most $\epsilon$. Therefore, by Lemma 3.3, $|\text{cov}[A(u, \sigma), A(v, \sigma)]| = o_\epsilon(1)$. \hfill $\square$

**Claim 3.5.** Suppose that $G_n$ has sfpg. Then $\text{Var}[/\epsilon(G_n)] = o(1)$.

**Proof.** For a vertex $w$, denote by $R_w$ the event that $w \in I(G_n)$. Let $u, v$ be two independently and uniformly chosen vertices from $G_n$. Since the random variables $\mathbb{E}[R_u \mid u]$ and $\mathbb{E}[R_v \mid v]$ are independent, by Claim 3.4,

\[
\text{Var}[/\epsilon(G_n)] = \mathbb{E}[\text{cov}[R_u, R_v \mid u, v]] \\
= \text{cov}[R_u, R_v] - \text{cov}[\mathbb{E}[R_u \mid u]\mathbb{E}[R_v \mid v]] = \text{cov}[R_u, R_v] = o(1). \hfill \square
\]
Proof of Theorem 1.3. Let \( \varepsilon > 0 \). Note that since \( G_n \) has \( \text{sfpG} \), \((U, \rho)\) has nonexplosive growth, hence by Theorem 1.1 there exists \( n_0 \) such that for every \( n \geq n_0 \), \(|\bar{\iota}(G_n) - \iota(U, \rho)| \leq \varepsilon \). Thus, by Chebyshev’s inequality and Claim 3.5,
\[
\mathbb{P}[|\iota(G_n) - \iota(U, \rho)| > 2\varepsilon] \leq \mathbb{P}[|\iota(G_n) - \bar{\iota}(G_n)| > \varepsilon] \leq \varepsilon^{-2} \text{Var}[\iota(G_n)] = \mathcal{O}(1).
\]

4 Branching processes and differential equations

As promised, we begin with a formal definition of multitype branching processes. Let \( T \) be a finite or countable set, which we call the type set. Let \( \bar{\mu} \) be a distribution on \( T \), which we call the root distribution, and for each \( k \in T \) let \((\mu^{k\rightarrow j})_{j \in T}\) be an offspring distribution, which is a distribution on vectors with nonnegative integer coordinates. Let \( \tau \sim \bar{\mu} \) and for every finite sequence of natural numbers \( v \) let \((\xi^{k\rightarrow j}_v)_{j \in T} \sim (\mu^{k\rightarrow j})_{j \in T}\) be a random vector, where these random vectors are independent for different indices \( v \) and are independent of \( \tau \).

A multitype branching process \((Z_t)_{t \in \mathbb{N}}\) with type set \( T \), root distribution \( \bar{\mu} \) and offspring distributions \((\mu^{k\rightarrow j})_{j \in T}\) is a Markov process on labelled trees, in which each vertex is assigned a type in \( T \), which may be described as follows. At time \( t = 0 \) the tree \( Z_0 \) consists of a single vertex of type \( \tau \), labelled by the empty sequence. At time \( t + 1 \) the tree \( Z_{t+1} \) is obtained from \( Z_t \) as follows. For each \( k \in T \) and \( v \) of length \( t \) and type \( k \) in \( Z_t \), we add the vertices \( v \sim i \) for all \( 0 \leq i < \sum_{j \in T} \xi^{k\rightarrow j}_v \), having exactly \( \xi^{k\rightarrow j}_v \) of them being assigned type \( j \), uniformly at random, and connecting them with edges to \( v \).

If in addition \((\mu^{k\rightarrow j})_{j \in T}\) is a product measure, namely, if \( \xi^{k\rightarrow j} \sim \mu^{k\rightarrow j} \) are sampled independently for distinct \( j \in T \), the process is called simple. We often think of a multitype branching process as the possibly infinite (random) rooted graph \( Z_\infty = \bigcup_{t \geq 0} Z_t \), rooted at the single vertex of \( Z_0 \).

Proof of Theorem 1.3. Let \( \sigma \) be a random labelling of \( U \). To ease notation, set \( \iota = \iota(U, \rho) \) and \( \mathbf{I} = \mathbf{I}(U[\mathcal{P}_\rho]) \), and recall that \( \iota = \mathbb{P}[\rho \in \mathbf{I}] \). Let \( \tau \sim \bar{\mu} \) be the type of the root. For \( k \in T \) and \( x \in [0, 1] \), define \( \iota^{(k)} = \mathbb{P}[\rho \in \mathbf{I} \mid \tau = k] \) and \( \iota^{(k)}_x = \mathbb{P}[\rho \in \mathbf{I} \mid \sigma_\rho = x, \tau = k] \). Note that this is well defined, even if the event that \( \sigma_\rho = x \) has probability 0. Let further
\[
\iota^{(k)}_{< x} = \int_0^x \iota^{(k)}_z \, dz,
\]
so \( \iota^{(k)} = \iota^{(k)}_{<1} \), hence
\[
\iota = \sum_{k \in T} \iota^{(k)}_{<1} \cdot \mathbb{P}[\tau = k].
\]

It therefore suffices to show that the family \( y_k(x) := \iota^{(k)}_{< x} \) satisfies (\( \ast \)) (it clearly satisfies the boundary conditions). The key observation is that distinct children in the past of the root are roots to independent subtrees. Formally, conditioning on the event that \( v_1, \ldots, v_a \) are the children of \( \rho \) in its past, the events “\( v_i \in \mathbf{I} \)” for \( i = 1, \ldots, a \) are mutually independent. Since \( \rho \in \mathbf{I} \) if and only if \( v_i \notin \mathbf{I} \) for every \( i = 1, \ldots, a \),
\[
y_k(x) = (\iota^{(k)}_{< x}) = \iota^{(k)}_x = \sum_{\ell \in \mathbb{N}^T, j \in T} \mu^{k\rightarrow j}_x(\ell_j)(1 - \mathbb{P}[\rho \in \mathbf{I} \mid \sigma_\rho < x, \tau = j])^{\ell_j}
\]
\[
= \sum_{\ell \in \mathbb{N}^T, j \in T} \mu^{k\rightarrow j}_x(\ell_j) \left(1 - \frac{y_j(x)}{x}\right)^{\ell_j}.
\]
\( \square \)

\( ^6 \)By \( v \sim i \) we mean the sequence obtained from \( v \) by appending the element \( i \).
5 Probability generating functions

The goal of this section is to demonstrate how generating functions may aid solving the fundamental system of ODEs (*) (and thus finding \( \iota \)) for certain simple branching processes. In the following sections, we will use the notation \( y_k(x) \) as in (*), and omit the subscript \( k \) when the branching process has a single type.

**Single Type Branching Processes** For a probability distribution \( p = (p_d)_{d=0}^\infty \), let \( T_p \) be the \( p \)-ary tree, namely, it is a (single type) branching process, for which the offspring distribution is \( p \). The fundamental ODE in this case is

\[
y'_{0}(x) = \sum_{d=0}^{\infty} p_d \sum_{\ell=0}^{d} \binom{d}{\ell} (1-x)^{d-\ell} x^\ell \left(1 - \frac{y(x)}{x}\right)^\ell = \sum_{d=0}^{\infty} p_d (1-y(x))^d.
\]

This differential equation may not be solvable, but in many important cases it is, and we will use it. Denote by \( g_p(z) \) the probability generating function (pgf) of \( p \), that is,

\[
g_p(z) = \sum_{d=0}^{\infty} p_d z^d.
\]

Let \( h_p(x) \) be the solution to the equation

\[
\int_{h_p(x)}^{1} \frac{dz}{g_p(z)} = x.
\]

**Claim 5.1.** \( y(x) = 1 - h_p(x) \).

**Proof.** Fix \( x \in [0,1] \), let \( h = h_p(x) \) and \( g(z) = g_p(z) \). Define \( \varphi : [0, \beta] \to [h,1] \), where \( \beta = y^{-1}(1-h) \), as follows: \( \varphi(u) = 1 - y(u) \). Note that by (2),

\[
\varphi'(u) = -y'(u) = -g(\varphi(u)).
\]

Thus

\[
x = \int_{h}^{1} \frac{dz}{g(z)} = \int_{\varphi(0)}^{\varphi(\beta)} \frac{dz}{g(z)} = \int_{0}^{\beta} \varphi'(z) dz = \beta,
\]

hence \( y(x) = 1 - h \).

In particular, it follows from Claim 5.1 that \( \iota(T_p) = 1 - h_p(1) \).

**Random Trees With IID Degrees** For a probability distribution \( p = (p_d)_{d=1}^\infty \), let \( T_p \) be the \( p \)-tree, namely, it is a random tree in which the degrees of the vertices are independent random variables with distribution \( p \). We may view it as a two-type branching process, with type 0 for the root and 1 for the rest of the vertices. Let \( g_p(z) \) be the pgf of \( p \) (see (3), and note that \( p_0 = 0 \)). The fundamental system of ODEs in this case is

\[
y'_{0}(x) = \sum_{d=1}^{\infty} p_d \sum_{\ell=0}^{d} \binom{d}{\ell} (1-x)^{d-\ell} x^\ell \left(1 - \frac{y_1(x)}{x}\right)^\ell = \sum_{d=1}^{\infty} p_d (1-y_1(x))^d = g_p(1-y_1(x)),
\]

11
and by (2),
\[ y_1'(x) = \sum_{d=0}^{\infty} p_{d+1}(1 - y_1(x))^d = \frac{1}{1 - y_1(x)} \sum_{d=1}^{\infty} p_d(1 - y_1(x))^d = \frac{g_p(1 - y_1(x))}{1 - y_1(x)}. \]  

Let \( h_p(x) \) be the solution to the equation
\[ \int_{h_p(x)}^{1} \frac{zdz}{g_p(z)} = x. \]

The next claim is [15, Theorem 1].

**Claim 5.2.** \( y_0(x) = \frac{1}{2}(1 - h_p^2(x)). \)

**Proof.** Fix \( x \in [0,1] \), let \( h = h_p(x) \) and \( g(z) = g_p(z) \). Define \( \varphi : [0, \beta] \to [h, 1] \), where \( \beta = y_1^{-1}(1 - h) \), as follows: \( \varphi(u) = 1 - y_1(u) \). Note that by (6),
\[ \varphi'(u) = -y_1'(u) = -\frac{g(\varphi(u))}{\varphi(u)}. \]
Thus
\[ x = \int_{h}^{1} \frac{zdz}{g(z)} = \int_{\varphi(0)}^{\varphi(\beta)} \frac{zdz}{g(z)} = -\int_{0}^{\beta} \frac{\varphi'(z)\varphi(z)dz}{g(\varphi(z))} = \beta, \]
hence \( y_1(x) = 1 - h \). From (5) and (6) it follows that \( y_0(x) = g(h) = y_1'(x) \cdot h = -h \cdot h' \), and since \( y_0(0) = 0 \) it follows that \( y_0(x) = \frac{1}{2}(1 - h^2) \).

In particular, it follows from Claim 5.2 that \( \iota(T_p) = \frac{1}{2}(1 - h_p^2(1)) \).

### 6 Applications

The goal of this section is to demonstrate the power of the introduced framework by finding \( \iota \) for several natural (random) graph sequences, via finding their local limit and solving its fundamental system of ODEs, as described in Theorem 1.3. In some cases, we may use probability generating functions, as described in Section 5, to ease calculations.

#### 6.1 Infinite-Ray Stars

For \( d \geq 1 \), let \( S_d \) be the **infinite-ray star** with \( d \) branches. Formally, the vertex set of \( S_d \) is \( \{(0,0)\} \cup \{(i,j) : i \in [d], j = 1,2,\ldots\} \), and \( (i,j) \sim (i',j') \) if \( |j - j'| = 1 \) and either \( i = i' \) or \( i' = 0 \). Note that \( S_1 = \mathbb{N} \) and \( S_2 = \mathbb{Z} \). This is a two-type branching process, with types \( d \) for the root and 1 for a branch vertex. The fundamental system of ODEs in this case is \( y_0'(x) = (1 - y_1(x))^d \), and for \( d = 1 \) we obtain the equation \( y_1' = 1 - y_1 \) of which the solution is \( y_1(x) = 1 - e^{-x} \). For \( d > 1 \) we obtain the equation \( y_1' = e^{-dx} \) of which the solution is \( y_d(x) = \frac{1}{d}(1 - e^{-dx}) \). Since \( \tau = d \) a.s., it follows that \( \iota(S_d) = y_d(1) = \zeta_d := \frac{1}{d}(1 - e^{-d}) \). In particular, \( \iota(\mathbb{N}) = 1 - e^{-1} \approx 0.6321\ldots \) and \( \iota(\mathbb{Z}) = \frac{1}{2}(1 - e^{-2}) \approx 0.4323\ldots \).

As \( \mathbb{N} \) is a single type branching process and \( \mathbb{Z} \) is a random tree with iid degrees, we may use the alternative approach for calculating \( \iota(\mathbb{N}) \) and \( \iota(\mathbb{Z}) \), as described in Section 5. Solving \( \int_{0}^{1} \frac{dz}{z} = 1 \) gives \( h = e^{-1} \), hence by Claim 5.1, \( \iota(\mathbb{N}) = 1 - e^{-1} \), and by Claim 5.2, \( \iota(\mathbb{Z}) = \frac{1}{2}(1 - e^{-2}) \).

\(^7\)In [15] the authors required that the the degrees of the tree are all at least 2; we do not require this here.
**Paths and cycles** The local limit of the sequences $P_n$ of paths and $C_n$ of cycles is clearly $\mathbb{Z}$ (rooted at an arbitrary vertex). It follows from the discussion above that $\nu(P_n), \nu(C_n) \sim \frac{1}{2}(1 - e^{-2})$ **whp**. This was already calculated by Flory [22] (who only considered the expected ratio) and independently by Page [44], and can be thought of as the discrete variant of Rényi’s parking constant (see [19]).

### 6.2 Poisson Galton–Watson Trees

A Poisson Galton–Watson tree $T_λ$ is a single type branching process with offspring distribution $\text{Pois}(\lambda)$ for some parameter $\lambda \in (0, \infty)$. The fundamental ODE in this case is $y'(x) = e^{-\lambda y(x)}$ (This can be calculated directly using (2)). The solution for this differential equation is $y(x) = \ln(1 + \lambda x)/\lambda$, hence $\nu(T_λ) = y(1) = \ln(1 + \lambda)/\lambda$. The same result can be obtained using the probability generating function of the Poisson distribution, as described in Section 5.

**Random graphs** Consider the binomial random graph $G(n, \lambda/n)$, which is the graph on $n$ vertices in which every pair of nodes is connected by an edge independently with probability $\lambda/n$. It is easy to check that it converges locally to $T_λ$ (see, e.g., [14]), hence $\nu(G(n, \lambda/n)) \sim \ln(1 + \lambda)/\lambda$ **whp**, recovering a known result (see [39]).

### 6.3 Size-Biased Poisson Galton–Watson Trees

For $0 < \lambda \leq 1$, a size-biased Poisson Galton–Watson tree $T_λ$ can be defined (see [38]) as a two-type simple branching process, with types $s$ (spine vertices) and $t$ (tree vertices), where a spine vertex has 1 spine child plus $\text{Pois}(\lambda)$ tree children, a tree vertex has $\text{Pois}(\lambda)$ tree children, and the root is a spine vertex (when $\lambda = 1$, this is sometimes called the skeleton tree). The fundamental system of ODEs in this case is

$$y'_t(x) = 2 \sum_{d=0}^{\infty} \frac{(\lambda x)^d}{e^{\lambda x}d!} \left(1 - \frac{y_t(x)}{x}\right)^d \left(1 - \frac{y(x)}{x}\right)^d + (1 - x) \sum_{d=0}^{\infty} \frac{(\lambda x)^d}{e^{\lambda x}d!} \left(1 - \frac{y_t(x)}{x}\right)^d \left(1 - \frac{y(x)}{x}\right)^d,$$

and from Section 6.2 we obtain $y_t(x) = \ln(1 + \lambda x)/\lambda$. Hence $y'_t(x) = (1 - y_t(x))/(1 + \lambda x)$, and the solution for that equation is $y_t(x) = 1 - \exp(-\ln(1 + \lambda x)/\lambda)$. Thus $\nu(T_λ) = y_t(1) = 1 - (1 + \lambda)^{-1/\lambda} = 1 - e^{-\nu(T_λ)}$. In particular, $\nu(T_1) = 1/2$.

**Uniform spanning trees** It is a classical (and beautiful) fact (see, e.g., [26, 35]) that if $T_n$ is a uniformly chosen random tree drawn from the set of $n^{n-2}$ trees on (labelled) $n$ vertices, then $T_n$ converges locally to $T_1$, hence $\nu(T_n) \sim 1/2$ **whp**. To the best of our knowledge, this intriguing fact was not previously known. Recently, after a conference version of this paper was published, Contat [12] proved a much stronger statement concerning the cardinality of the greedy independent set in uniform random trees, showing that it has essentially the same law as its complement. Nachmias and Peres [41] showed (see also [29]) that if $G_n$ is a sequence of finite, simple, connected regular graphs with degree tending to infinity, and $T_n$ is the uniform spanning tree of $G_n$, then $T_n$ converges locally to $T_1$. It follows that $\nu(T_n) \sim 1/2$ **whp** in this case as well.
Random functional digraphs It can be easily verified that the local limit of a random functional digraph $\tilde{G}_1(n)$ (the digraph on $n$ vertices whose edges are $(i,\pi(i))$ for a uniform random permutation $\pi$), with orientations ignored, is also $\tilde{T}_1$, hence $\iota(\tilde{G}_1) \sim 1/2$ whp.

Sparse random planar graphs Let $\mathcal{P}(n, \lambda)$ denote the uniform distribution over the set of (labelled) planar graphs on $n$ vertices with $\lambda n/2$ edges. According to a recent result by Kang and Missenath [32], if $\lambda \in (0, 1]$ then $\mathcal{P}(n, \lambda)$ converges locally to $\mathcal{T}_\lambda$, hence $\iota(\mathcal{P}(n, \lambda)) \sim \ln(1 + \lambda)/\lambda$ whp; and if $\lambda \in (1, 2]$ then $\mathcal{P}(n, \lambda)$ converges locally to $(\lambda - 1)\tilde{T}_1 + (2 - \lambda)\tilde{T}_1$, namely, to the random tree which is sampled from $\tilde{T}_1$ with probability $\lambda - 1$ and from $\tilde{T}_1$ with probability $2 - \lambda$. It follows that in this case, $\iota(\mathcal{P}(n, \lambda)) \sim (\lambda - 1)/2 + (2 - \lambda)\ln 2$ whp. Note that $\iota(\mathcal{P}(n, \lambda))$ is continuous for $\lambda \in (0, 2]$.

6.4 $d$-ary Trees

For $d > 1$, let $T_d$ be the $d$-ary tree. It may be viewed as a (single type) branching process. It thus immediately follows from (2) that $y'(x) = (1 - y(x))^d$. The solution for this differential equation is $y(x) = 1 - ((d - 1)x + 1)^{-1/(d-1)}$. It follows that $\iota(T_d) = y(1) = 1 - d^{-1/(d-1)}$. This fact also follows easily using the generating functions approach described in Section 5. A remarkable example is $\iota(T_2) = 1/2$.

6.5 Regular Trees

For $d \geq 3$, let $T_d$ be the $d$-regular tree. It may viewed as a two-type branching process with types $d$ for the root and $d - 1$ for the rest of the vertices. The fundamental system of ODEs in this case is $y'_d(x) = (1 - y_{d-1}(x))^d$, and from Section 6.4 we obtain $y_{d-1}(x) = 1 - ((d - 2)x + 1)^{-1/(d-2)}$. It follows that $y'_d(x) = ((d - 2)x + 1)^{-d/(d-2)}$, of which the solution is $y_d(x) = (1 - ((d - 2)x + 1)^{-2/(d-2)})/2$. Therefore,

$$\iota(T_d) = y_d(1) = \frac{1}{2} \left(1 - (d - 1)^{-2/(d-2)}\right).$$

As with $d$-ary trees, here again the generating functions approach works easily: the solution to $\int h(x) x^{d-1} dz = x$ is $h(x) = (1 - (2 - d)x)^{1/(2-d)}$, and the result follows from Claim 5.2. Remarkable examples include $\iota(T_3) = 3/8$ and $\iota(T_4) = 1/3$.

Random regular graphs Since the random regular graph $G(n, d)$ (a uniformly sampled graph from the set of all $d$-regular graphs on $n$ vertices, assuming $dn$ is even) converges locally to $\tilde{T}_d$ (see, e.g., [56]), the above result for this case is exactly [54, Theorem 4]. In fact, since any sequence of $d$-regular graphs with girth tending to infinity converges locally to $\tilde{T}_d$, we also recover [37, Theorem 2].

7 Hypergraphs

Recently (after a conference version of this paper was posted), Nie and Verstraëte [43] analysed the random greedy algorithm for producing maximal independent sets in $r$-uniform $d$-regular high-girth (linear) hypergraphs. In this section we show how to deduce their main results in our framework, assuming the girth tends to infinity.
It is not hard to check that the local limit of \( r \)-uniform \( d \)-regular hypergraphs with girth tending to infinity is the rooted infinite \( r \)-uniform \( d \)-regular loose hypertree (see Fig. 1), denoted here by \( T^r_d \) (this is just the hypergraph-analogue to the graph case described in Section 6.5). This hypertree may be viewed as a two-type branching process with types \( d \) for the root and \( d-1 \) for the rest of the vertices, where a type \( d \) vertex has \( d \) incident edges, on each there are \( r-1 \) additional vertices of type \( d-1 \), and a type \( d-1 \) vertex has \( d-1 \) incident edges, on each there are \( r-1 \) additional vertices of type \( d-1 \). The fundamental system of ODEs in this case is thus

\[
\begin{align*}
y'_d(x) &= (1 - y^r_{d-1}(x))^d, \quad (7) \\
y'_{d-1}(x) &= (1 - y^r_{d-1}(x))^{d-1}. \quad (8)
\end{align*}
\]

The second equation is separable, thus

\[
\int dx = \int \frac{y_{d-1}(x)}{(1 - y^r_{d-1}(x))^{d-1}} d\left(1 - y^r_{d-1}(x)\right) = \int \frac{dy}{(1 - y^r_{d-1}(x))^{d-1}} = \int \sum_{n=0}^{\infty} \binom{n + d - 2}{d - 2} y^{n(r-1)} dy
\]

The solution to (8) is \( y_{d-1}(x) = H^{-1}(x) \). As for (7), set \( u = H^{-1}(t) \) and then (by (8)),

\[
du = (1 - u^{r-1})^{d-1} dt,
\]

hence

\[
y_d(x) = \int_0^x (1 - y^r_{d-1}(t))^{d} dt = \int_0^x (1 - (H^{-1}(t))^{r-1})^{d} dt = \int_{H^{-1}(0)}^{H^{-1}(x)} \frac{(1 - u^{r-1})^{d}}{(1 - u^{r-1})^{d-1}} du
\]

\[
= \int_0^{H^{-1}(x)} (1 - u^{r-1}) du = H^{-1}(x) - \frac{(H^{-1}(x))^{r}}{r}.
\]
In particular, \( \iota(T_r) = y_d(1) = H^{-1}(1) - (H^{-1}(1))^r/r \), and together with a hypergraph analogue of Theorem 1.1 this recovers [43, Theorem 4], up to the error bounds. Using (a hypergraph analogue of) Theorem 1.2 we also obtain a nonquantitative (namely, without an explicit bound on the deviation from the mean) version of [43, Theorem 5]. The advantage of the above method is, however, its greater generality, as it allows for a wider range of assumptions on the underlying hypergraph.

8 Lower bound in tree sequences

Let us focus on tree sequences. How large can the expected greedy independent ratio be? How small can it be? The sequence of stars is a clear witness that the only possible asymptotic upper bound is the trivial one, namely 1. Apparently, the lower bound is not trivial. An immediate corollary of Theorems 1.1 and 1.4 is that a tight asymptotic lower bound is

\[
\iota(Z) = \frac{1 - e^{-2}}{2}
\]

(compare with [50]). The statement of Theorem 1.4 is, however, much stronger: paths achieve the exact (non-asymptotic) lower bound for the expected greedy independence ratio among the set of all trees of a given order.

To prove Theorem 1.4 we will need to first understand the behaviour of the greedy algorithm on the path.

For a graph \( G \) denote by \( i(G) \) the cardinality of its greedy independent set, and let \( \overline{i}(G) = E[i(G)] \). Let \( \alpha_n = i(P_n) \). Suppose the vertices of \( P_n \) are 1, \ldots, \( n \), and let \( S \) be the vertex which is first in the permutation of the vertices. Setting \( \alpha_{-1} = \alpha_0 = 0 \), we obtain the recursion

\[
\alpha_n = E[E[i(P_n) \mid S]] = \frac{1}{n} \sum_{i=1}^{n} (1 + \alpha_{i-2} + \alpha_{n-i-1}) = 1 + \frac{2}{n} \sum_{i=1}^{n} \alpha_{i-2}, \tag{9}
\]

and the following explicit formula for \( \alpha_n \) \((n \geq 0)\) appears in [23]:

\[
\alpha_n = \sum_{i=0}^{n-1} \frac{(-2)^i(n-i)}{(i+1)!}. \tag{10}
\]

We introduce the following notation to simplify the upcoming calculations. For a sequence \( x_n \) we write

\[
\Delta^d_{h} x_n = \sum_{k=0}^{d} \binom{d}{k} x_{n+(d-k)h}
\]

to denote its \( d^{th} \) order \( h \)-forward difference. When \( h = 1 \) we omit the subscript, and when \( d = 1 \) we omit the superscript.

**Lemma 8.1.** Fix \( k \geq 0 \). Let \( z_n \) be a real nonnegative decreasing sequence. Then \( y_{k,n} = (-1)^k \sum_{j=k}^{n} (-1)^j z_j \) is nonnegative.

**Proof.** Note that for every \( \ell \geq 0 \) we have that

\[
(-1)^{k+2\ell} z_{k+2\ell} \quad \text{and} \quad (-1)^{k+2\ell} z_{k+2\ell} + (-1)^{k+2\ell+1} z_{k+2\ell+1}
\]

are both nonnegative if \( k \) is even, and both nonpositive otherwise. The claim easily follows. \( \square \)

**Lemma 8.2.** Let \( z_n \) be a real nonnegative decreasing sequence, which is convex for \( n \geq 1 \). Then, the sequence \( x_n = \sum_{i=0}^{n-1} \sum_{j=0}^{i} (-1)^j z_j \) is monotone increasing and subadditive.
Proof. For $i, k \geq 0$, write $y_{k,i} = (-1)^k \sum_{j=k}^{k+i} (-1)^j z_j$. By Lemma 8.1 $y_{k,i} \geq 0$, hence $\Delta x_n = y_{0,n} \geq 0$, and $x_n$ is monotone increasing. Fix $m \geq 1$ and write $b_i = y_{i,m-1}$. We have that
\[
a_n := x_{m+n} - x_m - x_n = \sum_{i=0}^{m+n-1} y_{0,i} - \sum_{i=0}^{m-1} y_{0,i} - \sum_{i=0}^{n-1} y_{0,i}
= \sum_{i=0}^{n-1} (y_{0,m+i} - y_{0,i}) = \sum_{i=0}^{n-1} (-1)^{i+1} y_{i+1,m-1} = \sum_{i=1}^{n} (-1)^i b_i.
\]
Moreover,
\[
\Delta b_i = (-1)^{i+1} \sum_{j=i+1}^{i+m} (-1)^j z_j - (-1)^i \sum_{j=i}^{i+m-1} (-1)^j z_j
= (-1)^{i+1} \sum_{j=i}^{i+m-1} (-1)^j z_j + (-1)^{i+1} \sum_{j=i}^{i+m-1} (-1)^j z_j
= (-1)^{i+1} \sum_{j=i}^{i+m-1} (-1)^j (-\Delta z_j).
\]
Now, $-\Delta z_j$ is nonnegative, and for $j \geq 1$ it is also decreasing (since $\Delta^2 z_j \geq 0$), thus for $i \geq 1$, by Lemma 8.1, $\Delta b_i \leq 0$. Therefore, $b_i$ is nonnegative and decreasing (for $i \geq 1$), hence by Lemma 8.1, $a_n \leq 0$ for every $n \geq 0$, and thus $x_n$ is subadditive.

The following equations may be useful. Using (10),
\[
\Delta \alpha_n = \sum_{i=0}^{n} \frac{(-2)^i (n+1-i)}{(i+1)!} - \sum_{i=0}^{n-1} \frac{(-2)^i (n-i)}{(i+1)!} = \sum_{i=0}^{n} \frac{(-2)^i}{(i+1)!},
\]
and
\[
\Delta^2 \alpha_n = \frac{(-2)^n}{(n+2)!}.
\]
Note also that since $\alpha_0 = 0$,
\[
\alpha_n = \sum_{i=0}^{n-1} \Delta \alpha_i,
\]
and since $\Delta \alpha_0 = 1 = \Delta^2 \alpha_{-1}$,
\[
\Delta \alpha_i = \sum_{j=-1}^{i-1} \Delta^2 \alpha_j.
\]
We proceed by a (rather long) sequence of technical claims, which will be used later in the proof of Theorem 1.4, or, more concretely, in the proof of Theorem 8.12.
Define $\beta_n = (-1)^n \Delta^2 \alpha_{n-1}$.

Claim 8.3. $\beta_n$ is nonnegative and decreasing, and convex for $n \geq 1$.  

17
Proof. From (12) we know that  
\[ \beta_n = \frac{2^{n+1}}{(n+2)!} - \frac{2^n}{(n+1)!} = \frac{2^n}{(n+1)!} \left( \frac{2}{n+2} - 1 \right) \leq 0, \]
and, for \( n \geq 1, \)
\[ \Delta^2 \beta_n = \frac{2^{n+2}}{(n+3)!} - \frac{2 \cdot 2^{n+1}}{(n+2)!} + \frac{2^n}{(n+1)!} \]
\[ = \frac{2^n}{(n+1)!} \left( \frac{4}{(n+2)(n+3)} - \frac{4}{n+2} + 1 \right) \geq 0. \]

\[ \square \]

Claim 8.4. \( \alpha_n \) is monotone increasing and subadditive.

Proof. From (13) and (14) it follows that
\[ \alpha_n = \sum_{i=0}^{n-1} \Delta \alpha_i = \sum_{i=0}^{n-1} \sum_{j=-1}^{i-1} \Delta^2 \alpha_j = \sum_{i=0}^{n-1} \sum_{j=0}^{i} (-1)^j \beta_j, \]
and the result follows from Lemma 8.2 and Claim 8.3.

\[ \square \]

Define \( \gamma_n = (-1)^{n+1} \Delta \Delta \alpha_n. \)

Claim 8.5. \( \gamma_n \) is nonnegative and decreasing, and convex for \( n \geq 1. \)

Proof. Note that
\[ \gamma_n = (-1)^{n+1}(\Delta \alpha_{n+1} - \Delta \alpha_n) \]
\[ = (-1)^{n+1}(\alpha_{n+3} - \alpha_{n+2} - \alpha_{n+2} + \alpha_n) \]
\[ = (-1)^{n+1}(\Delta \alpha_{n+2} - \Delta \alpha_n) \]
\[ = (-1)^{n+1} \left( \frac{(-2)^{n+2}}{(n+3)!} + \frac{(-2)^{n+1}}{(n+2)!} \right) \]
\[ = \frac{2^{n+1}}{(n+2)!} \left( \frac{-2}{n+3} + 1 \right) > 0. \]
Moreover,
\[ \Delta \gamma_n = (-1)^{n+2}(\Delta \alpha_{n+2} - \Delta \alpha_{n+1}) - (-1)^{n+1}(\Delta \alpha_{n+1} - \Delta \alpha_n) \]
\[ = (-1)^{n}(\Delta \alpha_{n+2} - \Delta \alpha_{n+1}) \]
\[ = (-1)^{n}(\Delta \alpha_{n+3} + \Delta \alpha_{n+2} - \Delta \alpha_{n+1} - \Delta \alpha_n) \]
\[ = (-1)^{n} \left( \frac{(-2)^{n+3}}{(n+4)!} + 2 \cdot \frac{(-2)^{n+2}}{(n+3)!} + \frac{(-2)^{n+1}}{(n+2)!} \right) \]
\[ = \frac{2^{n+1}}{(n+2)!} \left( -\frac{4}{(n+3)(n+4)} + \frac{4}{n+3} - 1 \right) \leq 0, \]
so $\gamma_n$ is decreasing. Finally,
\[
\Delta^2 \gamma_n = \gamma_{n+2} - 2\gamma_{n+1} + \gamma_n \\
= (-1)^{n+1}(\Delta_2 \alpha_{n+3} + \Delta_2 \alpha_{n+2} - 2\alpha_{n+3} - 2\alpha_{n+2} + \alpha_{n+1} + \alpha_n) \\
= (-1)^{n+1}(\Delta_2 \alpha_{n+4} + 2\Delta_2 \alpha_{n+3} - 2\Delta_2 \alpha_{n+1} - \Delta_2 \alpha_n) \\
= (-1)^{n+1}\left(\frac{(2n+4)}{(n+5)!} + \frac{3\cdot(2n+3)}{(n+4)!} + \frac{3\cdot(2n+2)}{(n+3)!} + \frac{(2n+1)}{(n+2)!}\right) \\
= \frac{2^{n+1}}{(n+2)!}\left(-\frac{8}{(n+5)(n+4)(n+3)} + \frac{12}{(n+4)(n+3)} - \frac{6}{(n+3)} + 1\right),
\]
which is nonnegative for $n \geq 1$. \qed

For $a, b \geq 1$, define $\eta^{a,b}_n = (-1)^{n+1}\Delta_a \Delta_b \alpha_n$.

**Claim 8.6.** For every $a \geq 1$, $\eta^{a,2}_n$ is nonnegative.

**Proof.** Note that
\[
\eta^{a,2}_n = (-1)^{n+1}(\Delta_2 \alpha_{a+n} - \Delta_2 \alpha_n) \\
= (-1)^{n+1}\sum_{j=n}^{a+n-1} \Delta \Delta_2 \alpha_j = (-1)^{n}\sum_{j=n}^{a+n-1} (-1)^j \gamma_j,
\]
which is, by Lemma 8.1 and Claim 8.5, nonnegative. \qed

**Claim 8.7.** For every $a \geq 1$, $\eta^{a,1}_n$ is nonnegative and decreasing.

**Proof.** Note that
\[
\eta^{a,1}_n = (-1)^{n+1}(\Delta \alpha_{a+n} - \Delta \alpha_n) \\
= (-1)^{n+1}\sum_{j=n}^{a+n-1} \Delta^2 \alpha_j = (-1)^n\sum_{j=n}^{a+n-1} (-1)^j \Delta^2 \beta_{j+1},
\]
which is, by Lemma 8.1 and Claim 8.3, nonnegative. Moreover,
\[
\Delta \eta^{a,1}_n = (-1)^n(\Delta \alpha_{a+n+1} - \Delta \alpha_{n+1} + \Delta \alpha_{a+n} - \Delta \alpha_n) \\
= (-1)^n(\Delta_2 \alpha_{a+n} - \Delta_2 \alpha_n) = (-1)^n \Delta_2 \alpha_n = -\eta^{a,2}_n,
\]
which is, by Claim 8.6, nonpositive, hence $\eta^{a,1}_n$ is decreasing. \qed

Define $\psi^{b}_n = (-1)^{n+1}\Delta \Delta_b \alpha_{n+1} + \Delta \Delta_b \alpha_n$.

**Claim 8.8.** For every $b \geq 1$, $\psi^{b}_n$ is nonnegative, and decreasing for $n \geq 1$.

**Proof.** Note that
\[
\psi^{b}_n = (-1)^{n+1}(\Delta_b \alpha_{n+2} - \Delta_b \alpha_n) = (-1)^{n+1}\Delta_2 \Delta_b \alpha_n = (-1)^{n+1}\Delta_2 \Delta_b \alpha_n = \eta^{b,2}_n,
\]
which is, by Claim 8.6, nonnegative. Moreover,
\[
\Delta \psi^n_b = \eta^{b,2}_n - \eta^{b,2}_{n+1} = (-1)^{n+1} \sum_{j=n+1}^{b+n} (-1)^j \gamma_j - (-1)^n \sum_{j=n}^{b+n-1} (-1)^j \gamma_j
\]
\[
= (-1)^{n+1} \sum_{j=n}^{b+n-1} \left( (-1)^{j+1} \gamma_{j+1} + (-1)^j \gamma_j \right)
\]
\[
= (-1)^{n+1} \sum_{j=n}^{b+n-1} (-1)^{j+1} \Delta \gamma_j.
\]
By Claim 8.5, the sequence \(-\Delta \gamma_n\) is nonnegative, and decreasing for \(n \geq 1\). Therefore, by Lemma 8.1, \(\Delta \psi^n_b\) is nonpositive, thus \(\psi^n_b\) is decreasing (for \(n \geq 1\)).

**Claim 8.9.** For every \(a, b \geq 1\), \(\eta^{a,b}_n\) is nonnegative, and decreasing for \(n \geq 1\).

**Proof.** Note that
\[
\eta^{a,b}_n = (-1)^{n+1} (\Delta_b \alpha_{a+n} - \Delta_b \alpha_n)
\]
\[
= (-1)^{n+1} \sum_{j=n}^{a+n-1} \Delta_b \Delta \alpha_j
\]
\[
= (-1)^n \sum_{j=n}^{a+n-1} (-1)^j \eta^{b,1}_j,
\]
which is, by Lemma 8.1 and Claim 8.7, nonnegative. Moreover,
\[
\Delta \eta^{a,b}_n = (-1)^{n+2} (\Delta_b \alpha_{a+n+1} - \Delta_b \alpha_{n+1}) - (-1)^{n+1} (\Delta_b \alpha_{a+n} - \Delta_b \alpha_n)
\]
\[
= (-1)^n (\Delta_b \alpha_{a+n+1} - \Delta_b \alpha_{n+1} + \Delta_b \alpha_{a+n} - \Delta_b \alpha_n)
\]
\[
= (-1)^n \sum_{j=n}^{a+n-1} (\Delta \Delta_b \alpha_{j+1} + \Delta \Delta_b \alpha_j)
\]
\[
= (-1)^{n+1} \sum_{j=n}^{a+n-1} (-1)^{j} \psi^n_b,
\]
which is, for \(n \geq 1\), by Lemma 8.1 and Claim 8.8, nonpositive, hence \(\eta^{a,b}_n\) is decreasing (for \(n \geq 1\)).

Define \(\xi_{n,\ell} = \sum_{j=1}^\ell \alpha_{n+j}\).

**Claim 8.10.** For every \(\ell, a, b \geq 1\) it holds that \(\xi_{a,\ell} + \xi_{b,\ell} \leq \xi_{a+b,\ell} + \xi_{0,\ell}\).

**Proof.** Note that
\[
\xi_{a+b,\ell} + \xi_{0,\ell} - \xi_{a,\ell} - \xi_{b,\ell} = \sum_{j=1}^\ell (\alpha_{a+b+j} + \alpha_j - \alpha_{a+j} - \alpha_{b+j}) = -\sum_{j=1}^\ell (-1)^j \eta^{a,b}_j
\]
which is, by Lemma 8.1 and Claim 8.9, nonnegative. 

[20]
8.1 KC-Transformations

In this section we introduce the main tool that will be used to prove Theorem 1.4. Let $T$ be a tree and let $x, y$ be two distinct vertices of $T$. We say that the path between $x$ and $y$ is bare if for every vertex $v \neq x, y$ on that path, $d_T(v) = 2$. Suppose $x, y$ are such that the unique path $P$ in $T$ between them is bare, and let $z$ be the neighbour of $y$ in that path. For a vertex $v$, denote by $N(v)$ the neighbours of $v$ in $T$. The KC-transformation $KC(T, x, y)$ of $T$ with respect to $x, y$ is the tree obtained from $T$ by deleting every edge between $y$ and $N(y) \setminus z$ and adding the edges between $x$ and $N(y) \setminus z$ instead (see Fig. 2). Note that $KC(T, x, y) \cong KC(T, y, x)$, so if we only care about unlabelled trees, we may simply write $KC(T, P)$, for a bare path $P$ in $T$. The term “KC-transformation” was coined by Bollobás and Tyomkyn [10] after Kelmans, who defined a similar operation on graphs [34], and Csikvári, who defined it in this form [13] under the name “generalized tree shift” (GTS).

A nice property of KC-transformations, first observed by Csikvári [13], is that they induce a graded poset on the set of unlabelled trees of a given order, which is graded by the number of leaves. In particular, this means that in that poset, the path is the unique minimum (say) and the star is the unique maximum. Note that if $P$ contains a leaf then $KC(T, P) \simeq T$, and otherwise $KC(T, P)$ has one more leaf than $T$. In the latter case, we say that the transformation is proper.

Here is the plan for how to prove Theorem 1.4. For a tree $T$ and a vertex $v$, denote by $T * v$ the forest obtained from $T$ by shattering $T$ at $v$, that is, by removing from $T$ the set $\{v\} \cup N(v)$ (see Fig. 3). Denote by $\kappa_v(T)$ the multiset of orders of trees in the forest $T * v$, and by $\kappa(T)$ the sum of $\kappa_v(T)$ for all vertices $v$ in $T$. Note that for trees with up to 3 vertices,
Theorem 1.4 is trivial; we proceed by induction. By the induction hypothesis,
\[ \tilde{i}(T) = \frac{1}{n} \sum_{v \in V(T)} \left( 1 + \sum_{S \in T \ast v} \tilde{i}(S) \right) \geq 1 + \frac{1}{n} \sum_{v \in V(T)} \sum_{k \in \kappa(T)} \alpha_k = 1 + \frac{1}{n} \sum_{k \in \kappa(T)} \alpha_k. \] (15)
Therefore, it makes sense to study the quantities \( \nu(T) = \sum_{k \in \kappa(T)} \alpha_k \) and \( \nu(T) = \sum_{k \in \kappa(T)} \alpha_k \).
In fact, it would suffice to show that for any tree \( T \) on \( n \) vertices \( \nu(T) \geq \nu(P_n) \), since by (9) and (15) we would obtain
\[ \tilde{i}(T) \geq 1 + \frac{1}{n} \nu(T) \geq 1 + \frac{1}{n} \nu(P_n) = \tilde{i}(P_n). \]

We therefore reduced our problem to proving the following theorem about KC-transformations.

**Theorem 8.11.** If \( T \) is a tree and \( P \) is a bare path in \( T \) then \( \nu(KC(T, P)) \geq \nu(T) \).

It would have been nice if for every \( v \in V(T) \) we would have had \( \nu_v(KC(T, P)) \geq \nu_v(T) \); unfortunately, this is not true in general. However, the following statement would suffice.

**Theorem 8.12.** Let \( T \) be a tree and let \( x \neq y \) be two vertices with the path between them being bare. Denote \( T' = KC(T, x, y) \). Let \( A \) be the set of vertices \( v \neq x \) in \( T \) for which every path between \( v \) and \( y \) passes via \( x \), and similarly, let \( B \) be the set of vertices \( v \neq y \) in \( T \) for which every path between \( v \) and \( x \) passes via \( y \). Let \( P \) be the set of vertices on the bare path between \( x \) and \( y \), so \( A \cup B \cup P \) is a partition of \( V(T) \). Then
1. For \( v \in A \cup B \) we have that \( \nu_v(T') \geq \nu_v(T) \).
2. \( \sum_{v \in P} \nu_v(T') \geq \sum_{v \in P} \nu_v(T) \).

**Proof.**

1. It suffices to prove the claim for \( v \in A \). First note that there exists a unique tree \( S_v \) in \( T \ast v \) which is not fully contained in \( A \), and the rest of the trees are retained in the KC-transformation. The set of trees in \( T' \ast v \) which are not fully contained in \( A \) may be different from \( S_v \), but they are on the same vertex set, so the result follows from subadditivity of \( \alpha_n \) (Claim 8.4).

2. Write \( |A| = a \), \( |B| = b \) and \( |P| = \ell + 1 \). Let \( A_1, \ldots, A_a \) be the trees of \( T \ast x \) which are fully contained in \( A \), and denote \( a_i = |A_i| \). Let \( B_1, \ldots, B_b \) be the trees of \( T \ast y \) which are fully contained in \( B \), and denote \( b_i = |B_i| \). Let \( \alpha_A = \sum_{i=1}^{a} \alpha_{a_i} \), \( \alpha_A' = \sum_{i=1}^{a} \alpha_{a_i+1} \), \( \alpha_B = \sum_{i=1}^{b} \alpha_{b_i} \) and \( \alpha_B' = \sum_{i=1}^{b} \alpha_{b_i+1} \). Denote the vertices of \( P \) by \( x = u_0, u_1, \ldots, u_\ell \). The following table summarises the values of \( \nu \) in \( T, T' \) along vertices of \( P \), in the case where \( \ell \geq 3 \) (similar tables can be made for the cases \( \ell = 1, 2 \)).

<table>
<thead>
<tr>
<th></th>
<th>( \nu_{a_j}(T) )</th>
<th>( \nu_{a_j}(T') )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( j = 0 )</td>
<td>( \alpha_A + \alpha_{b+\ell-1} )</td>
<td>( \alpha_A + \alpha_B + \alpha_{\ell-1} )</td>
</tr>
<tr>
<td>( j = 1 )</td>
<td>( \alpha_A' + \alpha_{b+\ell-2} )</td>
<td>( \alpha_A' + \alpha_B' + \alpha_{\ell-2} )</td>
</tr>
<tr>
<td>( 2 \leq j \leq \ell - 2 )</td>
<td>( \alpha_{a+j-1} + \alpha_{b+\ell-j-1} )</td>
<td>( \alpha_{a+b+j-1} + \alpha_{\ell-j-1} )</td>
</tr>
<tr>
<td>( j = \ell - 1 )</td>
<td>( \alpha_{a+\ell-2} + \alpha_B' )</td>
<td>( \alpha_{a+b+\ell-2} )</td>
</tr>
<tr>
<td>( j = \ell )</td>
<td>( \alpha_{a+\ell-1} + \alpha_B )</td>
<td>( \alpha_{a+b+\ell-1} )</td>
</tr>
</tbody>
</table>
It follows (for every $\ell \geq 1$) that
\[
\sum_{v \in P} (\nu_v(T') - \nu_v(T)) = \sum_{j=1}^{\ell-1} (\alpha_{a+b+j} + \alpha_j - \alpha_{a+j} - \alpha_{b+j})
= \xi_{a+b,\ell-1} + \xi_{0,\ell-1} - \xi_{a,\ell-1} - \xi_{b,\ell-1},
\]
which is, by Claim 8.10, nonnegative.

\[\square\]

\section{9 Concluding remarks and open questions}

\textbf{Non locally tree-like graph sequences} Our local limit approach does not assume that the converging sequence is locally tree-like. However, the differential equation tool fails completely if short cycles appear in a typical local view. As it seems, to date, there is no general tool to handle these cases, and indeed, even the asymptotic behaviour of the random greedy MIS algorithm on $d$-dimensional tori (for $d \geq 2$) remains unknown.

\textbf{Better local rules} The random greedy algorithm presented here follows a very simple local rule. More complicated local rules may yield, in some cases, larger maximal independent sets; for example, the initial random ordering may “favour” low degree vertices. It would be nice to adapt our framework, or at least some of its components, to other settings. For adaptive “better” local algorithms we refer the reader to [54, 57].

\textbf{The second colour} In this work we have analysed the output of the random greedy algorithm for producing a maximal independent set. As already remarked, this is in fact the set of vertices in the first colour class in the random greedy colouring algorithm. It is rather easy to see, that, after slight modifications (in particular, in Theorem 1.3) this approach allow us to calculate the asymptotic proportion of the size of the set of vertices in the second colour class (or in the $k$'th colour class in general, for any fixed $k$) as well. Non-asymptotic questions about the expected cardinality of the set of vertices in the second colour class might be also of interest. For example, is it true that the path has the smallest expected number of vertices in the first two colour classes among all trees of the same order? It is not hard to see that this statement is not true for the first three colour classes (as three colours suffice to greedily colour the path).

\textbf{Monotonicity with respect to KC-transformations} It is likely that the expected greedy independence ratio in trees is monotone with respect to KC-transformations, and strictly monotone with respect to proper KC-transformations. If true, this would imply that the greedy independence ratio in trees achieves its unique minimum on the path and its unique maximum on the star.

\section*{References}


24


