

## ON THE ORDER OF THE LARGEST INDUCED TREE IN A RANDOM GRAPH

Zbigniew PALKA and Andrzej RUCIŃSKI

*Institute of Mathematics, A. Mickiewicz University, Matejki 48/49, Poznań, Poland*

Received 27 September 1984

Revised 24 January 1986

Consider a random graph  $K(n, p)$  with  $n$  labeled vertices in which the edges are chosen independently and with a probability  $p$ . Let  $T_n(p)$  be the order of the largest induced tree in  $K(n, p)$ . Among other results it is shown, using an algorithmic approach, that if  $p = (c \log n)/n$ , where  $c \geq e$  is a constant, then for any fixed  $\varepsilon > 0$

$$\left(\frac{1}{c} - \varepsilon\right) \frac{\log \log n}{\log n} n < T_n(p) < \left(\frac{2}{c} + \varepsilon\right) \frac{\log \log n}{\log n} n$$

almost surely.

### 1. Introduction

Let  $\Omega$  be the family of all spanning subgraphs of a complete graph  $K_n$ . Denote by  $\mathcal{F}$  the power set of  $\Omega$  and define a probability measure on the discrete space  $(\Omega, \mathcal{F})$  as follows: for every graph  $G \in \Omega$

$$\text{Prob}(G) = p^t (1-p)^{\binom{n}{2}-t}$$

where  $t$  denotes the number of edges of the graph  $G$  and  $0 \leq p \leq 1$ . An element from  $\Omega$  is denoted by  $K(n, p)$  and called a *random graph*. We say that  $K(n, p)$  has a certain property  $\pi$  *almost surely* (a.s.) if

$$\text{Prob}(K(n, p) \text{ has property } \pi) \rightarrow 1 \quad \text{as } n \rightarrow \infty.$$

Let  $T_n = T_n(p)$  be the order of the largest induced tree in a random graph  $K(n, p)$ . It was shown in [2] that if the edge probability  $p$  is *fixed* (i.e.  $p$  does not depend on  $n$ ) then the sequence  $\{T_n\}$  of random variables satisfies

$$\frac{T_n}{\log n} \rightarrow \frac{2}{\log 1/q} \quad \text{as } n \rightarrow \infty$$

in probability. At the same time it was proved (see [6]) that (1.1) holds with probability one. (For a generalization of this result to a wider class of induced subgraphs see [8].) On the other hand, if  $p = p(n) = 1/n$ , then (see [3])

$$\frac{n^{2/3}}{\omega(n)} \leq T_n(p) \leq n^{2/3} \omega(n) \quad \text{a.s.}$$

where  $\omega(n)$  is a sequence tending to infinity (arbitrarily slowly) as  $n \rightarrow \infty$ . In [2] the following open problem was set. Find such a value of the edge probability  $p$  for which the random variable  $T_n(p)$  has the maximum value. It was conjectured there that if  $p = p(n) = c/n$ , where  $c > 1$  is a constant, then there exists  $\varphi(c) > 0$ , independent of  $n$ , such that  $T_n \geq \varphi(c)n$  a.s. Although we are not able to prove the above conjecture yet, we will look at this problem from an algorithmic point of view. An algorithmic approach was already used by a great many authors when investigating the independence number, chromatic number or tree number of a random graph (see e.g. [1], [4]–[7], [10]).

In this paper we describe a very simple greedy algorithm which for some specific values of the edge probability  $p$  constructs pretty large induced trees of  $K(n, p)$ . Among other results, we show that if  $p = (e \log n)/n$ , then for any fixed  $\varepsilon > 0$

$$T_n(p) > \left(\frac{1}{e} - \varepsilon\right) \frac{\log \log n}{\log n} n \quad \text{a.s.}$$

This is the best lower bound of  $T_n(p)$  obtained until this time.

As usual, for any real  $x$ ,  $\lfloor x \rfloor$  and  $\lceil x \rceil$  denote the greatest integer not greater than  $x$  and the least integer not less than  $x$ , respectively. The symbols  $o$  and  $O$  are used with respect to  $n \rightarrow \infty$ . Also, logarithms are to base  $e$ .

## 2. Algorithm

We begin with the description of a simple, but sometimes an impressive greedy algorithm for finding an induced tree in a given graph. Let  $G$  be any simple graph with vertex set  $\{1, 2, \dots, n\}$ . The algorithm TREE runs through the vertices in the order  $\{1, 2, \dots\}$  and selects a new vertex whenever it can be selected, i.e. whenever it is joined with exactly one vertex from the vertices selected so far. Note that vertex 1 always belongs to the constructed subgraph.

### Algorithm TREE

```

begin
   $F := \{1\}$ 
  for  $i = 2$  to  $n$  do
    if  $F \cup \{i\}$  is an induced tree
      then  $F := F \cup \{i\}$ 
end

```

Let us apply the algorithm TREE to a random graph  $K(n, p)$ . In order to make a precise probabilistic analysis of this algorithm we shall change slightly the model of our random graph. (We use the same approach as in e.g. [4], [5] or [7]).

Let  $\Omega^*$  be the family of all spanning subgraphs of an infinite complete graph on vertex set  $\mathbb{N} = \{1, 2, 3, \dots\}$ . If  $H \in \Omega^*$  and  $I \subseteq \mathbb{N}$ , we write  $H(I)$  for the subgraph of

$H$  induced by the vertices  $I$ . For each finite subset  $I \subseteq \mathbb{N}$  and each graph  $G$  with vertex set  $I$ , let

$$[G : I] = \{H \in \Omega^* : H(I) \text{ is } G\}.$$

That is,  $[G : I]$  is the subset of  $\Omega^*$  consisting of all members of  $\Omega^*$  which have  $G$  as their subgraph induced by  $I$ . The set of finite-dimensional cylinders of  $\Omega^*$  is the set of all such  $[G : I]$  as  $G$  ranges over all graphs on finite subsets  $I$  on  $\mathbb{N}$ . Let  $\mathcal{A}$  be the smallest  $\sigma$ -algebra of subsets of  $\Omega^*$  which contains the finite-dimensional cylinders of  $\Omega^*$ . We define a probability measure on  $(\Omega^*, \mathcal{A})$  by specifying its value on each finite-dimensional cylinder as follows:

$$\text{Prob}[G : I] = p^t (1-p)^{\binom{s}{2} - t}$$

where  $s$  and  $t$  are the number of vertices and edges in the finite graph  $G$ , respectively. If  $I = \{1, 2, \dots, n\}$  then we write  $K^*(n, p)$  for such defined random graph. It is clear that  $K^*(n, p)$  has the same probabilistic structure as  $K(n, p)$ . For this reason the results which will be proved for  $K^*(n, p)$  will certainly hold for  $K(n, p)$ .

Now we are ready to make a probabilistic analysis of the algorithm TREE. Let  $T_n^* = T_n^*(p)$  be the order of an induced tree in  $K^*(n, p)$  constructed by the algorithm. Define a function  $\varrho_k : \Omega^* \rightarrow \{0, 1, 2, \dots\}$  as follows:  $\varrho_0 = 0$  and for  $k \geq 1$

$$\varrho_k = \min\{s : \text{after the } s\text{th iteration the algorithm has constructed an induced tree of order } k\}.$$

Then  $\delta_k = \varrho_{k+1} - \varrho_k$  ( $k = 0, 1, 2, \dots$ ) defines a sequence of independent random variables with  $\delta_0 = 1$  and the  $\delta_k$  ( $k \geq 1$ ) distributed geometrically, namely

$$\text{Prob}(\delta_k = j) = (1 - p_k)^{j-1} p_k \quad (j = 1, 2, 3, \dots)$$

where  $p_k = kp(1-p)^{k-1}$ . Furthermore

$$\varrho_j = \sum_{k=0}^{j-1} \delta_k. \tag{2.1}$$

Notice also that the geometric random variable  $\delta_k$  ( $k \geq 1$ ) has the mean  $p_k^{-1}$  and variance  $(1 - p_k)p_k^{-2}$ . Consequently, by (2.1) and the independence of  $\delta_k$ 's we have

$$E(\varrho_j) = 1 + \sum_{k=1}^{j-1} p_k^{-1} \tag{2.2}$$

and

$$\text{Var}(\varrho_j) = \sum_{k=1}^{j-1} (1 - p_k)p_k^{-2}. \tag{2.3}$$

Now, using the Chebyshev's inequality we obtain

$$\begin{aligned} \text{Prob}(\varrho_j > n) &\leq \text{Prob}(|\varrho_j - E(\varrho_j)| \geq n - E(\varrho_j)) \\ &\leq \frac{\text{Var}(\varrho_j)}{(n - E(\varrho_j))^2} \end{aligned} \tag{2.4}$$

if  $E(\varrho_j) < n$  and analogously

$$\text{Prob}(\varrho_j \leq n) \leq \frac{\text{Var}(\varrho_j)}{(E(\varrho_j) - n)^2} \quad (2.5)$$

if  $E(\varrho_j) > n$ . These two inequalities together with the following obvious relation

$$\text{Prob}(T_n^* < j) = \text{Prob}(\varrho_j > n) \quad (2.6)$$

are the principal tools in proving our main results which are presented in the next section.

### 3. Results

We will give a probabilistic analysis of the algorithm TREE with respect to different values of the edge probability  $p = p(n)$ . As we mentioned in the introduction for some specific values of  $p$  the algorithm constructs very large induced trees. On the other hand, it is interesting that sometimes our algorithm can not construct even an induced tree of a small order although it is known that a random graph does contain a large tree (compare Theorem 3.1(a) below with (1.2)). The following result shows that the algorithm TREE is very uneffective for all edge probabilities  $p$  such that  $d/n \leq p \leq (c \log n)/n$ , where  $d > 0$  and  $0 < c < 1$  are constants.

**Theorem 3.1.** (a) *If  $p = d/n$ , where  $d > 0$  is a constant, then for any  $\varepsilon > 0$  there exists a constant  $a = a(\varepsilon)$  such that*

$$\text{Prob}(T_n^*(p) \geq a(\xi)) \leq \varepsilon.$$

(b) *If  $p = \psi(n)/n$ , where  $\omega(n) \rightarrow \infty$  in such a way that  $\omega(n) \leq c \log n$  and  $0 < c < 1$  is a constant, then for arbitrarily small  $\varepsilon > 0$*

$$\text{Prob}(T_n^*(p) \geq \exp[(1 + \varepsilon)\psi(n)]) = o(1).$$

**Proof.** Since the method of the proof is the same in both cases we will show here only the second part of the theorem. Using the left-hand side of the inequality (see [9, p. 18])

$$\log \frac{N+1}{N-i+2} \leq \sum_{k=N-i+2}^N \frac{1}{k} \leq \log \frac{N}{N-i+1} \quad (3.1)$$

where  $i = N - m$  and  $m \geq 0$  is a natural number, we obtain by (2.2)

$$\begin{aligned} E(\varrho_j) &> \frac{n}{\omega(n)} \sum_{k=1}^{j-1} \frac{1}{k} \\ &\geq \frac{n}{\omega(n)} (1 + \log(j/2)) \\ &> \frac{n}{\omega(n)} \log j. \end{aligned}$$

On the other hand, by (2.3) and the relation

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6} \quad (3.2)$$

we have

$$\begin{aligned} \text{Var}(\varrho_j) &\leq \frac{n^2}{\omega(n)_2} (1-p)^{-2j} \sum_{k=1}^{j-1} k^{-2} \\ &= O\left(\frac{n^2}{\omega(n)^2} \exp\left(\frac{2j\omega(n)}{n}\right)\right). \end{aligned}$$

Consequently, if  $j = \lceil \exp[(1+\varepsilon)\omega(n)] \rceil$ , then  $E(\varrho_j) > (1+\varepsilon)n$  and by (2.5)

$$\text{Prob}(\varrho_j \leq n) = O\left(\psi(n)^{-2} \exp\left[\frac{2\omega(n)}{n} e^{(1+\varepsilon)\psi(n)}\right]\right) = o(1)$$

provided  $0 < \varepsilon < (1/c) - 1$ . Thus taking the complementary events in (2.6) we get our results.  $\square$

A radical change of the effectiveness of the algorithm TREE takes place when the edge probability  $p$  reaches the value of  $(\log n)/n$ . For the sake of simplicity let us put

$$f(n) = \frac{\log \log n}{\log n}. \quad (3.3)$$

The following result is true.

**Theorem 3.2.** *If  $p = (\log n)/n$ , then for any fixed  $\varepsilon > 0$*

$$\text{Prob}(T_n^*(p) \geq (2-\varepsilon)n f(n)^2) = 1 - O((\log \log n)^{-2}).$$

**Proof.** Let  $j = \lfloor (2-\varepsilon)n f(n)^2 \rfloor$  and  $i = \lfloor j/\log \log n \rfloor$ . Applying (2.2) and the right-hand side of the inequality (3.1) we have

$$\begin{aligned} E(\varrho_j) &= 1 + \frac{n}{\log n} \left\{ \sum_{k=1}^i \frac{(1-p)^{-k+1}}{k} + \sum_{k=i+1}^{j-1} \frac{(1-p)^{-k+1}}{k} \right\} \\ &\leq 1 + \frac{n}{\log n} \left\{ \exp\left[(2-\varepsilon)f(n) + O\left(\frac{\log \log n}{n}\right)\right] (1 + \log i) \right. \\ &\quad \left. + \exp\left[(2-\varepsilon)\frac{(\log \log n)^2}{\log n} + O\left(\frac{(\log \log n)^2}{n}\right)\right] \log \log \log n \right\} \\ &= n \exp\left[(2-\varepsilon)f(n) - 2f(n) + O\left(\frac{\log \log \log n}{\log n}\right)\right] + o(n) \\ &= n \left\{ 1 - \varepsilon f(n) + O\left(\frac{\log \log \log n}{\log n}\right) \right\}. \end{aligned}$$

Furthermore, similarly as in the proof of Theorem 3.1, we obtain

$$\text{Var}(\varrho_j) = O\left(\frac{n^2}{(\log n)^2} \exp\left[\frac{(\log \log n)^2}{\log n}\right]\right) = O\left(\frac{n^2}{(\log n)^2}\right).$$

Consequently, by (2.4) and (2.6) we have

$$\text{Prob}(T_n^* < j) = O((\log \log n)^{-2})$$

which completes the proof.  $\square$

The best result for  $T_n^*(p)$  we are able to show is in the case when  $p = (c \log n)/n$ ,  $c > 1$ . We have

**Theorem 3.3.** *Let  $p = (c \log n)/n$ ,  $c > 1$  a constant and let  $f(n)$  be defined by (3.3).*

(a) *If  $1 \leq c \leq e$ , then for any  $\varepsilon > 0$*

$$\text{Prob}\left(\left(\frac{\log c}{c} - \varepsilon\right)nf(n) < T_n^*(p) < \left(\frac{1}{c} + \varepsilon\right)nf(n)\right) = 1 - O((\log n)^{-2c\varepsilon}).$$

(b) *If  $c \geq e$ , then for any  $\varepsilon > 0$*

$$\text{Prob}\left(\left|\frac{T_n^*(p)}{nf(n)} - \frac{1}{c}\right| < \varepsilon\right) = 1 - O((\log n)^{-2c\varepsilon})$$

*i.e.*

$$\frac{T_n^*(p)}{nf(n)} \rightarrow \frac{1}{c} \text{ as } n \rightarrow \infty$$

*in probability.*

**Proof.** Let  $c > 1$  and  $a = \min\{\log c, 1\}$ . For an arbitrary small  $\varepsilon > 0$  let us put

$$j = \left\lfloor \left(\frac{a}{c} - \varepsilon\right)nf(n) \right\rfloor \quad \text{and} \quad i = \lfloor j/\log \log n \rfloor.$$

Proceeding analogously as in the proof of Theorem 3.2 we obtain

$$E(\varrho_j) \leq \frac{n}{c} e^{a - c\varepsilon} (1 + o(1))$$

and

$$\text{Var}(\varrho_j) = O(n^2(\log n)^{-2c\varepsilon}).$$

Since for any  $c > 1$ ,  $e^{a - c\varepsilon} < c$ , so by (2.4) and (2.6) we have

$$\text{Prob}\left(T_n^*(p) < \left\lfloor \left(\frac{a}{c} - \varepsilon\right)nf(n) \right\rfloor\right) = O((\log n)^{-2c\varepsilon}). \quad (3.4)$$

On the other hand, if we put

$$j = \left\lceil \left( \frac{1}{c} + \varepsilon \right) n f(n) \right\rceil \quad \text{and} \quad i = \lceil j/A \rceil,$$

where  $A = A(\varepsilon)$  is a constant such that  $1 < A < 1 + c\varepsilon$ , then by (2.2) and the left-hand side of (3.1) we have

$$\begin{aligned} E(q_j) &> \frac{n}{c \log n} \sum_{k=i}^{j-1} \frac{1}{k} (1-p)^{-k+1} \\ &\geq \frac{n}{c \log n} \exp(pj/A) \log A \\ &= \frac{\log A}{c} n (\log n)^{(1+c\varepsilon)/A-1}. \end{aligned}$$

Furthermore, by (2.3),

$$\text{Var}(q_j) \leq \left( \frac{n}{c \log n} \right)^2 \left\{ \sum_{k=1}^{j/2} k^{-2} (1-p)^{-2k} + \sum_{k=j/2+1}^j k^{-2} (1-p)^{-2k} \right\}.$$

Thus, taking into account (3.2) and the relation

$$\sum_{k=N+1}^{\infty} k^{-2} = O\left(\frac{1}{N}\right)$$

(see [9, p. 19]) we obtain

$$\begin{aligned} \text{Var}(q_j) &\leq \left( \frac{n}{c \log n} \right)^2 \left\{ (1-p)^{-j} \frac{\pi^2}{6} + (1-p)^{-2j} \sum_{k=j/2+1}^{\infty} k^{-2} \right\} \\ &= O\left( \left( \frac{n}{\log n} \right)^2 \exp[(1+c\varepsilon) \log \log n] \right. \\ &\quad \left. + \frac{n}{(\log n)(\log \log n)} \exp[(2+2c\varepsilon) \log \log n] \right) \\ &= O(n^2 (\log n)^{c\varepsilon-1}). \end{aligned}$$

Since  $A < 1 + c\varepsilon$ , we can use (2.5) and finally by (2.6)

$$\begin{aligned} \text{Prob} \left( T_n^*(p) \geq \left\lceil \left( \frac{1}{c} + \varepsilon \right) n f(n) \right\rceil \right) &= O((\log n)^{c\varepsilon+1-2(1+c\varepsilon)/A}) \\ &= O((\log n)^{-2c\varepsilon}) \end{aligned}$$

if only  $A < (2+2c\varepsilon)/(1+3c\varepsilon)$ . But such a constant always exists, since for any  $0 < \varepsilon < 1/3c$  we have  $1 + c\varepsilon < (2+2c\varepsilon)/(1+3c\varepsilon)$ . Consequently, by (3.4) and (3.5) we deduce our result.  $\square$

From the second part of the last theorem we see that the order of the largest induced tree in a random graph  $K(n, p)$ , where  $p = (e \log n)/n$  satisfies

$$T_n(p) > \left(\frac{1}{e} - \varepsilon\right) \frac{\log \log n}{\log n} n \quad \text{a.s.}$$

This is the best lower bound of  $T_n(p)$  obtained until this time. As usual, it is interesting to know the difference between the order of an induced tree constructed by the algorithm TREE and the order of the largest induced tree which in fact exists in a random graph. It appears that if  $p = (c \log n)/n$  where  $c > 1$ , then  $T_n^*(p)$  differs from  $T_n(p)$  only by a constant. As a matter of fact, the following result holds.

**Theorem 3.4.** *Let  $p = (c \log n)/n$  where  $c > 1$  is a constant. Then for any fixed  $\varepsilon > 0$*

$$\text{Prob}\left(T_n(p) \geq \left(\frac{2}{c} + \varepsilon\right) n f(n)\right) = o(1).$$

**Proof.** It suffices to show that the expected value  $E(X_k)$  of the number of induced trees of order

$$k = \left\lceil \left(\frac{2}{c} + \varepsilon\right) n f(n) \right\rceil$$

tends to zero as  $n \rightarrow \infty$ . But by the Stirling's formula we have

$$\begin{aligned} E(X_k) &= \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{\binom{k}{2} - (k-1)} \\ &= \frac{n}{c \log n} (2\pi k^5)^{-1/2} \left\{ (c \log n) \exp \left[ 1 - \frac{ck \log n}{2n} + O(kp^2) \right] \right\}^k \\ &= O\left(\frac{n}{\log n} k^{-5/2} [c(\log n)^{-c\varepsilon/2}]^k\right) = o(1). \quad \square \end{aligned}$$

From the last two results we deduce the following

**Corollary 3.5.** *Let  $p = (c \log n)/n$  where  $c \geq e$ . Then for any fixed  $\varepsilon > 0$*

$$\left(\frac{1}{c} - \varepsilon\right) \frac{\log \log n}{\log n} n < T_n(p) < \left(\frac{2}{c} + \varepsilon\right) \frac{\log \log n}{\log n} n \quad \text{a.s.} \quad \square$$

## References

- [1] B. Bollobás and A. Thomason, Random graphs of small order, *Annals Discr. Math.* 28 (1985) 47–97.
- [2] P. Erdős and Z. Palka, Trees in random graphs, *Discr. Math.* 46 (1983) 145–150; Addendum: *Ibid.* 48 (1984) 331.



- [3] P. Erdős and A. Rényi, On the evolution of random graphs, *Publ. Math. Inst. Hungar. Acad. Sci.* 5 (1960) 17–61.
- [4] P. Gazmuri, Independent sets in random sparse graphs, *Networks* 14 (1984) 367–377.
- [5] G.R. Grimmett and C.J.H. McDiarmid, On colouring random graphs, *Math. Proc. Cambridge Philos. Soc.* 77 (1975) 313–324.
- [6] A. Marchetti-Spaccamela and M. Protasi, The largest tree in a random graph, *Theoret. Comput. Sci.* 23 (1983) 273–286.
- [7] C.J.H. McDiarmid, Colouring random graphs, *Annals of O.R.I.* (1984) 183–200.
- [8] A. Ruciński, Induced subgraphs of a random graph, to appear.
- [9] V.F. Kolchin, B.A. Sevast'yanov and V.P. Chistyakov, *Random Allocations* (Nauka, Moskva, 1976). (In Russian.)
- [10] W.F. de la Vega, Random graphs almost optimally colorable in polynomial time, *Annals Discr. Math.* 28 (1985) 311–317.

### Note added in proof

The conjecture of Erdős and Palka [2] that for  $p = c/n$ ,  $c > 1$  is a constant, there exists  $\varphi(c) > 0$  such that

$$T_n(p) \geq \varphi(c)n \quad \text{a.s.}$$

was confirmed independently by Frieze and Jackson (“Large induced trees in sparse random graphs” – submitted), Kučera (personal communication) and De la Vega (“Induced trees in sparse random graphs” – submitted). They applied more sophisticated algorithms than the algorithm TREE presented in this paper.