Embracing the giant component

Abraham Flaxman* David Gamarnik† Gregory B. Sorkin†

Abstract

Consider a game in which edges of a graph are provided a pair at a time, and the player selects one edge from each pair, attempting to construct a graph with a component as large as possible. This game is in the spirit of recent papers on *avoiding* a giant component, but here we *embrace* it.

We analyze this game in the offline and online setting, for arbitrary and random instances, which provides for interesting comparisons. For arbitrary instances, we find a large lower bound on the competitive ratio. For some random instances we find a similar lower bound holds with high probability (**whp**). If the instance has $\frac{1}{4}(1+\epsilon)n$ random edge pairs, when $0 < \epsilon \leq 0.003$ then any online algorithm generates a component of size $O((\log n)^{3/2})$ **whp**, while the optimal offline solution contains a component of size $\Omega(n)$ **whp**. For other random instances we find the average-case competitive ratio is much better than the worst-case bound. If the instance has $\frac{1}{2}(1-\epsilon)n$ random edge pairs, with $0 < \epsilon \leq 0.015$, we give an online algorithm which finds a component of size $\Omega(n)$ **whp**.

1 Introduction

A pair of recent papers [BF01, BFW02] analyze the "Achlioptas process", where a collection of random edge pairs is given a pair at a time, and the object is to select one edge from each pair to *avoid* having a (suitably defined) giant component in the resulting graph. Without any intelligent selection process, a giant component forms after about $\frac{1}{2}n$ edges; [BF01] shows that a strategy exists which accepts at least $0.535n$ edges without forming a giant component; [BFW02] shows (among other things) that no more than about 0.964 edges may be accepted.

It is equally natural to ask the opposite question,

> What can you do to encourage a random graph to form
> a giant component, using fewer than $(1+\epsilon)n/2$ edges?

In fact, it is so natural we learned that Bohman and Kravitz are studying it independently [BK03].

We now define the problem of Embracing the Giant Component (EGC) more precisely. An instance I consists of a sequence of m pairs of edges on n vertices. (If you like, I may be regarded as an

*Department of Mathematical Sciences, Carnegie Mellon University
†Department of Mathematical Sciences, IBM T.J. Watson Research Center

element of $\left[\binom{n}{2}\right]^{2m}$.) Edges, including those in a pair, may or may not be distinct. A solution is a choice of one edge from each pair, and its value is the order (number of vertices) in the largest component in the graph consisting of the chosen edges. EGC(I) is the maximum value of a solution for instance I.

We focus on *online* versions of EGC, in which we see the pairs one at a time and must select our edge before seeing the next pair, but we also consider *offline* versions, in which we see all m pairs before making our choice. In either case, we consider edge pairs chosen *randomly* (defining an average-case behavior) or arbitrarily (chosen adversarially).

In addition to being a natural graph-game problem, EGC has two other sources of interest. First, imagine that you are a company trying to build up a network of some sort, each new link you build must be in response to a customer demand, and your budget allows you to spend at a rate which satisfies only half of all new requests. Presuming that a large connected component in the network is beneficial to your customers and to you, your goal is to solve an optimization problem very similar to EGC. Of course any real-world problem would be much more complicated, with different costs and benefits for different links, the ability to wait longer or shorter times to see more choices, and so forth, but it is conceivable that there are real-world problems whose mathematical core is captured by EGC.

The second motivation is that EGC provides an example of a problem for which the competitive ratio is awful in the worst case, but, for certain parameters, quite reasonable in an average case; a previous example was given by [SSS02]. For certain other parameters, EGC has a lower bound on average-case competitive ratio that is almost as awful as in the worst case.

1.1 Worst Case

We first observe that in the worst case, it is hard to solve offline EGC exactly (to select edges giving a component as large as possible), or even to approximate it to better than some fixed factor.

Theorem 1 *Offline EGC is MAX SNP-hard.*

In the online setting, it is natural to measure performance in terms of the *competitive ratio*, the ratio $z_{\text{opt}}/z_{\text{online}}$ between the sizes of the components produced by the best possible offline and online algorithms. The next theorem shows that in the worst case, the competitive ratio is as bad as it conceivably could be.

Theorem 2 *The worst-case competitive ratio for EGC is* $(m+1)/2$. *Specifically, for every online algorithm, there is a sequence of* m *edge pairs for which the algorithm produces a collection of isolated edges, yet the optimal solution has a component on* $m+1$ *vertices.*

As we remark after the proof of this theorem, a competitive-ratio lower bound of $\Omega(n/\log n)$ holds even for randomized online algorithms against an oblivious adversary.

1.2 Average case

We define $I_{n,m}$ to be a *random* instance of EGC in which each edge of each pair is chosen independently, uniformly at random from the edge set of the complete graph K_n.

Our main intention is to compare the average-case competitive ratio with the worst-case lower bound in Theorem 2. To do so, we need some idea of the optimal offline value of $\mathrm{EGC}(I_{n,m})$. We will see that these random instances exhibit a sharp threshold in objective value at $m = \frac{1}{4}n$, which we will prove by analyzing a greedy heuristic for offline EGC.

Throughout the paper, we will rely on a "component-identification algorithm". This algorithm, and our method of analysis, is quite standard in the random-graph literature; see for example the giant-component chapter of *Random Graphs* [JŁR00, pp. 108–111].

Our component-identification algorithm, Algorithm $\mathcal{A}$, maintains two set of vertices, called *unborn*, U_i, and *alive*, A_i. Initially, a single vertex is alive, $A_1 = \{v_1\}$, and the remainder are unborn, $U_1 = [n] \setminus \{v_1\}$. At step i, we look at all the neighbors of some vertex $v_i \in A_i$. We kill v_i and give birth to all its unborn neighbors (formally, let $P_i = U_i \cap N(v_i)$ be the *progeny* of v_i, and set $A_{i+1} = A_i \setminus \{v_i\} \cup P_i$ and $U_{i+1} = U_i \setminus P_i$).

Our greedy heuristic is very similar to Algorithm $\mathcal{A}$. Roughly, we try starting at each vertex, and using the first edge we see from each pair. We will elaborate on this description in the proof of Theorem 3.

Theorem 3 *For any fixed $\epsilon > 0$, for $m = \frac{1}{4}(1-\epsilon)n$, we have $\mathrm{EGC}(I_{n,m}) = O(\log n)$ while for $m = \frac{1}{4}(1+\epsilon)n$, our greedy heuristic finds a solution showing $\mathrm{EGC}(I_{n,m}) = \Omega(n)$* ***whp****.*

The below-the-threshold half of the theorem follows from well-known results in the theory of random graphs, since the union of all the edges in all the pairs is a random graph with $\frac{1}{2}(1-\epsilon)n$ edges, which is below the threshold for a giant component (see, for example, [JŁR00]).

It is interesting to note that below the threshold, the largest component in the union of the edges contains at most one edge from each pair **whp**, so for $m = \frac{1}{4}(1-\epsilon)n$, we can solve $\mathrm{EGC}(I_{n,m})$ optimally **whp**.

The above-the-threshold half of the theorem is proved in Section 3, in a manner similar to the analysis of the giant component above the threshold in $G_{n,p}$.

Our next theorem shows that even in the average case, any online algorithm performs much worse than offline.

Theorem 4 *For $\epsilon \leq 0.003$, on instances $I_{n,m}$ with $m = \frac{1}{4}(1+\epsilon)n$, every online algorithm finds a component of size only $O((\log n)^{3/2})$* ***whp****.*

Theorem 3 and 4 together give a lower bound on the *average-case competitive ratio* for EGC: the ratio of offline solution to online is $\Omega(n/(\log n)^{3/2})$ **whp**. This shows that the lower bound on competitive ratio for EGC is more robust than Theorem 2 alone indicates.

Theorem 4 is only true for some range of ϵ, however. For example, if $\epsilon > 1$, then taking the first edge from each pair yields a random graph above the giant component threshold, and so this trivial algorithm has a constant competitive ratio. We go slightly beyond the trivial bound in the next theorem.

Consider Algorithm $\mathcal{C}$, which does the following: for some γ to be determined later, for the first γn choices we take the first edge of each pair. For the remaining $m - \gamma n$ choices, we take the first edge unless it touches an isolated vertex, in which case we take the second edge.

Theorem 5 *For $\epsilon \leq 0.015$, on instances $I_{n,m}$ with $m = \frac{1}{2}(1-\epsilon)n$, Algorithm $\mathcal{C}$ yields a component of size $\Omega(n)$* ***whp****.*

2 Proofs of Worst-Case Theorems

Proof of Theorem 1: To show the hardness of approximating EGC, we reduce from MAX 3SAT-5. MAX 3SAT-5 is a structured relative of MAX 3-SAT, introduced by Feige, where every variable appears in exactly 5 clauses and a variable does not appear in a clause more than once. Feige proves that there is some $\epsilon > 0$ for which it is NP-hard to distinguish a satisfiable instance from an instance with at most $(1-\epsilon)m$ satisfiable clauses [Fei98].

Given a MAX 3SAT-5 instance, we make a EGC instance by including a vertex for each literal, ℓ, 3 vertices for each clause C_1, C_2, C_3, and an additional "root" vertex, r. We model the assignment by n edge pairs which decide if each variable is true or false: let pair i be $(\{r, x_i\}, \{r, \overline{x_i}\})$. We include $3m$ additional pairs: for the i-th literal, ℓ, of each clause C, include a pair of the form $(\{\ell, C_i\}, \{C_i, C_{i+1}\})$. If the assignment is satisfiable, there is a way of selecting edges which yields a component of size $3m + n + 1$. On the other hand, any selection of edges from the first n pairs corresponds naturally to some assignment. If a literal is not selected, then since it appears in at most 5 clauses and is not connected to the root, it can be in a component of size at most 16. Since it is NP-hard to distinguish satisfiable 3SAT-5 instances from instances with at most $(1-\epsilon)m$ clauses satisfiable, it is also NP-hard to distinguish instances of EGC with a component of size $\frac{8}{5}m + 1$ from those with a component of size at most $(\frac{8}{5} - \epsilon)m + 1$. □

Proof of Theorem 2: We will present a sequence of edge pairs, depending on the previous choices of the algorithm. The edge pairs will all come from a complete binary tree, and the edges in each pair will be siblings, i.e., of the form $(\{x, y\}, \{x, y'\})$. Whatever the algorithm chooses at step i — and for a fixed deterministic algorithm this choice is predictable — we make it wish it chose otherwise. So, if the algorithm selects edge $\{x, y\}$, the next pair we give it is $(\{y', z\}, \{y', z'\})$.

Thus, the online algorithm obtains a graph with only isolated edges, while making the opposite choice at every step would yield a component with m edges. □

Of course, the same $(m+1)/2$ ratio also applies to *randomized* online algorithm, if the adversary is allowed to see the algorithm's choice before constructing the next pair. Even if the adversary is required to fix a sequence of pairs in advance, and even if she does not know what randomized algorithm is being used, there is an almost equally bad instance. It is given by a random path down the tree and the siblings of the path edges, each edge paired with its sibling, the pairs presented in order from root to leaf. At each step, the online algorithm has probability only 1/2 of choosing the path edge rather than its sibling, and hence **whp** gets a largest component of size only $O(\log n)$.

3 Proofs of Average-Case Theorems

Proof of Theorem 3: We repeat more formally the greedy heuristic sketched in the introduction, in a form conducive to analysis. Algorithm $\mathcal{B}$ repeats the following n times, starting with each possible vertex for v_1. At each step, we maintain two sets of vertices, called *unborn*, U_i; and *alive*, A_i. Initially, a single vertex is alive, $A_1 = \{v_1\}$, and the remainder are unborn, $U_1 = [n] \setminus \{v_1\}$. At step i, we choose some vertex $v_i \in A_i$ and identify all previously unidentified pairs with an edge incident to v_i. For each such edge pair, we use the edge incident to v_i. We let $P_i = N(v_i) \cap U_i$ denote the set of newly discovered vertices, and we set $A_{i+1} = (A_i \setminus \{v_i\}) \cup P_i$ and $U_{i+1} = U_i \setminus P_i$.

For analysis, it is convenient to work with an instance resembling $G_{n,p}$. Let $I_{n,p}$ be a random instance formed by including each pair of edges independently with probability p. Thus, our probability space is $\{0,1\}^{\binom{n}{2}^2}$ with the product measure. We will show that the threshold value is $p = \frac{1}{n^3}$, which has expected $\frac{n}{4}$ pairs. We do so by analyzing the behavior of Algorithm $\mathcal{B}$ on $I_{n,(1+\epsilon)n^{-3}}$, which proceeds in two claims. We will then translate this result to random instances $I_{n,m}$ where $m = \frac{1}{4}(1+\epsilon)n$.

Let $p = (1+\epsilon)n^{-3}$, and let $\beta, \delta > 0$ be such that $(1+\epsilon)(1-\beta)^3 = 1+\delta$ and let $t_0 = 8(1+\delta)\delta^{-2}\log n$ and $t_1 = \beta n$.

Claim 1: Running Algorithm $\mathcal{B}$ on $I_{n,p}$ with any starting vertex v_1, either the algorithm halts before step t_0 or for all $t_0 \le t \le t_1$ we have $|A_t| \ge 1$ **whp**.

For this it is sufficient, for each t with $t_0 \le t \le t_1$, to identify t vertices of the component. Before we have identified a size βn component, there are at least $(1-\beta)n$ unborn vertices. So there are at least $2(n(1-\beta))\binom{(1-\beta)n}{2} \approx (n(1-\beta))^3$ candidate edge pairs which contribute a unique vertex to P_i. Thus we have

$$\mathbb{P}\left[\sum_{i=1}^{t} |P_i| \le t\right] \le \mathbb{P}\left[\sum_{i=1}^{t} \mathrm{B}((n(1-\beta))^3, p) \le t\right].$$

We also have

$$\mathbb{E}\left[\sum_{i=1}^{t} \mathrm{B}(n^3(1-\beta)^3, p)\right] = (1+\epsilon)(1-\beta)^3 t = (1+\delta)t,$$

so we use a standard Chernoff bound to show the probability that Algorithm $\mathcal{B}$, starting at any

vertex, halts at time t for any $t_0 \le t \le t_1$ is at most

$$\begin{aligned} n\sum_{t=t_0}^{t_1} \mathbb{P}\Big[\sum_{i=1}^{t} \mathrm{B}\left(n^3(1-\beta)^3, p\right) \le t\Big] &\le n\sum_{t=t_0}^{t_1} \exp\left(\frac{-(\delta t)^2}{2(1+\delta)t}\right) \\ &\le nt_1 e^{-\delta^2 t_0/2(1+\delta)} \\ &\le n^{-2}. \end{aligned}$$

Claim 2: There is some vertex v so that starting Algorithm $\mathcal{B}$ on v yields a component of size at least t_0 **whp**.

For this, we start Algorithm $\mathcal{B}$ on some vertex v, and if it fails to discover t_0 vertices, we start it on an unexplored vertex, v', and keep going. Each run, we expose at most t_0 vertices, so if we fail t_0 times, the number of edges exposed at each step dominates $\mathrm{B}(n^3(1-\beta)^3, p)$. Now, for Algorithm $\mathcal{B}$ to fail in every run, we must have that the total number of vertices exposed is less than the number of steps. But

$$\mathbb{P}\Big[\sum_{i=1}^{t_0^2} \mathrm{B}(n^3(1-\beta)^3, p) \le t_0^2\Big] \le e^{-\delta^2 t_0^2/2(1+\delta)} = o(n^{-2}).$$

Therefore, we have some vertex where Algorithm $\mathcal{B}$ runs for at least t_0 steps with probability $1 - o(n^{-2})$.

Claims 1 and 2 imply Algorithm $\mathcal{B}$ finds a component of size $t_1 = \beta n$ in $I_{n,p}$ **whp**.

To translate this result from $I_{n,p}$ to $I_{n,m}$, note that the probability $I_{n,p}$ has exactly $\frac{n^4}{4}p = \frac{1}{4}(1+\epsilon)n =: m$ edge pairs is

$$\mathbb{P}_{I_{n,p}}[\mathcal{M}] = \binom{\frac{n^4}{4}}{m} p^m (1-p)^{\frac{n^4}{4}-m} = O(n^{-1/2}),$$

where $\mathcal{M}$ denotes the event that I has m distinct edge pairs. Also note that the probability $I_{n,m}$ consists of m distinct edge pairs is

$$\mathbb{P}_{I_{n,m}}[\mathcal{M}] = \prod_{i=0}^{m-1} \left(1 - \frac{i}{\binom{n}{2}^2}\right) = 1 - O(n^{-2}).$$

And, by symmetry, for any particular instance $I^\star$ we have

$$\mathbb{P}_{I_{n,m}}[I_{n,m} = I^\star \,|\, \mathcal{M}] = \mathbb{P}_{I_{n,p}}[I_{n,p} = I^\star \,|\, \mathcal{M}].$$

So the probability of any event $\mathcal{E}$ in the $I_{n,m}$ model is related to the probability in the $I_{n,p}$ model by

$$\begin{aligned} \mathbb{P}_{I_{n,m}}[\mathcal{E}] &= \mathbb{P}_{I_{n,m}}[\mathcal{E} \,|\, \mathcal{M}]\mathbb{P}_{I_{n,m}}[\mathcal{M}] + \mathbb{P}_{I_{n,m}}[\mathcal{E} \,|\, \overline{\mathcal{M}}]\mathbb{P}_{I_{n,m}}[\overline{\mathcal{M}}] \\ &\le \mathbb{P}_{I_{n,m}}[\mathcal{E} \,|\, \mathcal{M}] + O(n^{-2}) \\ &= \mathbb{P}_{I_{n,p}}[\mathcal{E} \,|\, \mathcal{M}] + O(n^{-2}) \\ &= O(n^{1/2})\mathbb{P}_{I_{n,p}}[\mathcal{M}]\mathbb{P}_{I_{n,p}}[\mathcal{E} \,|\, \mathcal{M}] + O(n^{-2}) \\ &\le O(n^{1/2})\mathbb{P}_{I_{n,p}}[\mathcal{E}] + O(n^{-2}). \end{aligned}$$

Since the failure probability was $O(n^{-2})$ in the $I_{n,p}$ model, it is $O(n^{-3/2})$ for $I_{n,m}$. □

Proof of Theorem 4: We will analyze a wider class of algorithms. Instead of requiring the algorithm to choose edges at each step of the process, we will generate the first γn pairs, and allow the process to keep any components in the union with at least 2 edges, and additionally to keep up to γn of the isolated edges. Then we will generously allow the process to keep all edges from an additional $\frac{1}{4}(1+\epsilon)n - \gamma n$ pairs. A nonrigorous intuition for our proof is that the first γn pairs are "pretty much" isolated edges, and so the graph resulting from this process "looks like" the union of $\gamma n + 2(\frac{1}{4}(1+\epsilon) - \gamma)n = \frac{1}{2}(1+\epsilon-2\gamma)n$ random edges. For $\gamma > \frac{1}{2}\epsilon$ such a graph is below the threshold for the giant component.

This heuristic argument does not translate directly into a rigorous proof because the union of the edges in the γn pairs is not a collection of isolated edges. To work around this, we will bound the contribution of the components of 3 or more vertices in the union of the first γn pairs. Note that, by symmetry, it makes no difference which γn isolated edges the algorithm selects, so in this wider class of algorithms, the results of any selection process are the same. To prove the theorem, we decompose the graph into two parts. Let G' be the union of γn isolated edges and the components containing at least 3 vertices in the union of $2\gamma n$ random edges. Let G'' be the union of $2(\frac{1}{4}(1+\epsilon) - \gamma)n = \frac{1}{2}(1+\epsilon-4\gamma)n$ edges. We show that **whp** $G' \cup G''$ contains no component of size exceeding

$$t_1 = \delta^{-1} 6(1-\gamma)^{-2}(\log n)^{3/2}.$$

To simplify calculations, we make G'' a realization of $G_{n,p}$, with $p = (1+\epsilon-4\gamma)/n$. We will translate our results to $G_{n,m}$ at the end of the proof. Let $\epsilon, \gamma, \delta > 0$ so that

$$(1+\epsilon-4\gamma)\left(e^{-4\gamma} + 4\gamma e^{-8\gamma} + 2\gamma + \frac{(4\gamma)^2 e^{3(1-4\gamma)}}{1-4\gamma e^{1-4\gamma}}\right) = 1 - 2\delta.$$

Note that such ϵ, γ, δ exist, for example taking $\epsilon = 0.003, \gamma = 0.003$ and $\delta \approx 0.003177$.

Let T_k denote the number of components in G' with k vertices. Given the values of the T_k's, we use an exposure procedure similar to Algorithm $\mathcal{A}$ to prove $G' \cup G''$ has no large components. We expose all the vertices adjacent to v_i in G'' and if we discover a vertex of a connected component of G', we add every vertex of this component to the set A_{i+1}. Thus, at each step, and conditioned on any history, the size of P_i is stochastically dominated by

$$\sum_{k=1}^{n} k\,\mathrm{B}(kT_k, p),$$

and the probability that we discover a component of size exceeding t_1 is bounded by

$$\mathbb{P}\left[\sum_{i=1}^{t_1}\sum_{k=1}^{n} |P_i| \le t_1\right] \le \mathbb{P}\left[\sum_{i=1}^{t_1}\sum_{k=1}^{n} k\,\mathrm{B}(kT_k, p) \ge t_1\right].$$

Let $\mathcal{E}_1$ denote the event that G' contains no component with more than $K = 6(1-\gamma)^{-2}\log n$ vertices. Standard arguments show $\mathbb{P}[\overline{\mathcal{E}_1}] \le O(n^{-2})$ (see, for example, [JŁR00]). If $\mathcal{E}_1$ holds, we

need only consider the sum of weighted Binomial r.v.'s up to the K-th term. In other words, $\mathcal{E}_1$ implies

$$\sum_{k=1}^{n} k\,\mathrm{B}(kT_k, p) = \sum_{k=1}^{K} k\,\mathrm{B}(kT_k, p).$$

Let $Z = \sum_{k=1}^{K} k^2 T_k p$. Note that Z is the expectation of the sum above conditioned on the T_k's. Also note that the value of Z is dependent on G' only. We now obtain a bound on Z that holds **whp**.

We use a tree census results for sparse random graphs. It is known that in $G_{n,m=cn/2}$ we have the following: (see, for example, Pittel, [Pit90])

$$\mathbb{E}[T_k] \sim n(k^{k-2}c^{k-1}e^{-ck}/k!),$$

which, in G' applies to T_k with $k \geq 3$. We also have $T_2 \leq \gamma n$, and $\mathbb{E}[T_1] = e^{-4\gamma}n + 4\gamma e^{-8\gamma}n - 2T_2$. So we have

$$\begin{aligned}
\mathbb{E}[Z] &= \mathbb{E}\left[\sum_{k=1}^{K} k^2 T_k p\right] \\
&\leq (1+\epsilon-4\gamma)\left(e^{-4\gamma} + 4\gamma e^{-8\gamma} + 2\frac{T_2}{n} + \sum_{k=3}^{K} k^k (4\gamma)^{k-1} e^{-4\gamma k}/k!\right) \\
&\leq (1+\epsilon-4\gamma)\left(e^{-4\gamma} + 4\gamma e^{-8\gamma} + 2\gamma + (4\gamma)^{-1}\sum_{k=3}^{\infty}\left(4\gamma e^{1-4\gamma}\right)^k\right) \\
&= (1+\epsilon-4\gamma)\left(e^{-4\gamma} + 4\gamma e^{-8\gamma} + 2\gamma + \frac{(4\gamma)^2 e^{3(1-4\gamma)}}{1-4\gamma e^{1-4\gamma}}\right) \\
&= 1 - 2\delta.
\end{aligned}$$

Let $\mathcal{E}_2$ be the event that $Z \leq 1-\delta$. We use a form of the Azuma-Hoeffding inequality due to McDiarmid (see [Hoe63, McD89]) to show $\mathcal{E}_2$ holds **whp**. Note that changing one edge of G' can create or destroy at most two components in G'. So this can change the value of Z by at most $2K^2p$. Therefore,

$$\begin{aligned}
\mathbb{P}[\overline{\mathcal{E}_2}] &= \mathbb{P}[Z \geq 1-\delta] \\
&\leq \mathbb{P}[Z \geq \mathbb{E}[Z] + \delta] \\
&\leq \exp\left(-\frac{2\delta^2}{(2\gamma n)(2K^2p)^2}\right) \\
&\leq \exp\left(-\frac{\delta^2 n}{4K^4}\right).
\end{aligned}$$

Conditioning on $\mathcal{E}_1$ and $\mathcal{E}_2$, we have

$$\sum_{i=1}^{t_1}\sum_{k=1}^{n} k\,\mathrm{B}(kT_k, p) = \sum_{i=1}^{t_1}\sum_{k=1}^{K} k\,\mathrm{B}(kT_k, p), \tag{1}$$

and

$$\mathbb{E}\Big[\sum_{i=1}^{t_1}\sum_{k=1}^{K} k\,\mathrm{B}(kT_k,p)\Big] = Zt_1 \le (1-\delta)t_1.$$

To bound the probability that sum (1) is larger than t_1, we use the following Chernoff bound, from [AS00, Theorem A.1.18].

Theorem 6 *Let X_i, $1 \le i \le n$ be independent random variables with each $\mathbb{E}[X_i] = 0$ and no two values of any X_i ever more than one apart. Set $S = X_1 + \cdots + X_n$. Then $\mathbb{P}[S > a] < \exp(-2a^2)$.*

Applying this to (1), we have

$$\begin{aligned}
\mathbb{P}\Big[\sum_{i=1}^{t_1}\sum_{k=1}^{n} k\,\mathrm{B}(kT_k,p) \ge t_1 \,\Big|\, \mathcal{E}_1,\mathcal{E}_2\Big] &\le \mathbb{P}\Big[\sum_{i=1}^{t_1}\sum_{k=1}^{K} k\,\mathrm{B}(kT_k,p) \ge Zt_1 + \delta t_1\Big] \\
&\le \mathbb{P}\Big[\sum_{i=1}^{t_1}\sum_{k=1}^{K}\sum_{j=1}^{kT_k} \frac{(\mathrm{Be}(p)-p)k}{K} \ge \delta t_1/K\Big] \\
&\le \exp\left(-2\left(\delta t_1/K\right)^2\right) \\
&= n^{-2}.
\end{aligned}$$

So the probability there exists any vertex on which we run for at least t_1 steps is at most n^{-1} by the union bound.

Since $\mathbb{P}[\overline{\mathcal{E}_1}] + \mathbb{P}[\overline{\mathcal{E}_2}] \le o(n^{-1})$, we complete the theorem by observing that $G_{n,p}$ has at least $\frac{1}{2}(1+\epsilon-4\gamma)n$ edges with constant probability, and extra edges can only increase the size of the largest component, so our claim also holds in $G_{n,m}$. □

Proof of Theorem 5: We bound the size of components formed by this process by exposing edges starting from a vertex v_1 and tracking the number of vertices unborn and alive.

Consider decomposing the graph into G', the edges selected before time γn, and G'', the edges selected after.

To simplify calculations, we take G' to be a realization of $G_{n,p}$, with $p = 2\gamma/n$. Also, we generate G'' by applying our selection rule to a realization of $I_{n,p'}$, with $p' = 2(1-\epsilon-2\gamma)n^{-3}$ (recall this is an instance where every pair of edges is included independently with probability p'). Thus the expected number of edges in $G' \cup G''$ is $\gamma n + \frac{1}{2}(1-\epsilon-2\gamma)n = \frac{1}{2}(1-\epsilon)n$, as it should be.

Let $\alpha, \beta, \gamma, \delta, \epsilon, \eta, \theta > 0$ be such that $(1-\beta)(2\gamma) = 2\gamma - \epsilon/2 + \delta$, and

$$(1-\beta-(1+\eta)e^{-2\gamma})(1-\beta)^2/2 + (1-\beta-(1+\eta)e^{-2\gamma})(1-\delta)e^{-2\gamma}(1-\beta) = \alpha$$

and

$$\alpha 2(1-\epsilon-2\gamma) = 1 - 2\gamma - \epsilon/2 + \theta.$$

Note that such parameters exist, for example $\beta = \eta = 10^{-6}$, $\gamma = 0.4$, $\epsilon = 0.015$, yielding $\alpha \approx 0.521$, $\delta \approx 0.007$, and $\theta \approx 0.0002$. Let $t_0 = 8\max\{\delta^{-2}(1-\beta)\gamma, \theta^{-2}(1-\epsilon-2\gamma)\}\log n$ and $t_1 = \beta n$.

We wish to bound the probability Algorithm $\mathcal{C}$ halts with a component of size $t_0 \le t \le t_1$. For this, it is sufficient to bound the probability $\sum_{i=1}^{t} |P_i| \le t$ for all $t_0 \le t \le t_1$. We decompose P_i into $P_i = P_i' \cup P_i''$, where P_i' are the progeny contributed by edges in G' and P_i'' are the progeny contributed by edges in G'' (and not by edges in G'). If $\sum_{i=1}^{t} |P_i| \le t$, then either $\sum_{i=1}^{t} |P_i'| \le (2\gamma - \epsilon/2)t$ or $\sum_{i=1}^{t} |P_i''| \le (1 - 2\gamma + \epsilon/2)t$.

Now, at any step t, conditioned on any history that has not yet discovered βn vertices, we have $|P_i'|$ stochastically dominates $\mathrm{B}((1-\beta)n, p)$ and

$$\begin{aligned}\mathbb{P}\Big[\sum_{i=1}^{t} |P_i'| \le (2\gamma - \epsilon/2)t\Big] &\le \mathbb{P}\Big[\sum_{i=1}^{t} \mathrm{B}((1-\beta)n, p) \le (2\gamma - \epsilon/2)t\Big] \\ &\le e^{-\delta^2 t/4\gamma(1-\beta)} \\ &\le n^{-2}.\end{aligned}$$

Let $\mathcal{E}_3$ denote the event that G' contains $(1\pm\eta)e^{-2\gamma}n$ isolated vertices. We omit a simple calculation using Chebyschev's inequality to show $\mathcal{E}_3$ holds **whp**.

Conditioning on $\mathcal{E}_3$, we have that at any step t, conditioned on any history that has not yet discovered βn vertices, there are at least $(1-\beta-(1+\eta)e^{-2\gamma})n$ vertices that are not isolated in G' and are still in U_i. So there are at least $((1-\beta-(1+\eta)e^{-2\gamma})(1-\beta)^2/2 + (1-\beta-(1+\eta)e^{-2\gamma})(1-\delta)e^{-2\gamma}(1-\beta))n^3 = \alpha n^3$ unexposed edge pairs which would cause our selection rule to place a vertex in P_i'. So

$$\begin{aligned}\mathbb{P}\Big[\sum_{i=1}^{t} |P_i''| \le (1 - 2\gamma - \epsilon/2)t\Big] &\le \mathbb{P}\Big[\sum_{i=1}^{t} \mathrm{B}(\alpha n^3, p') \le (1 - 2\gamma - \epsilon/2)t\Big] \\ &\le e^{-\theta^2 t/4\alpha(1-\epsilon-2\gamma)} \le n^{-4}.\end{aligned}$$

Thus by the union bound, the probability that some component has size t for $t_0 \le t \le t_1$ is $O(n^{-2})$.

Now, as in the proof of Theorem 3, we argue that some component has size at least t_0 **whp**. The argument is identical to the earlier theorem, and the size of the progeny at each stage is bounded identically to the previous paragraph, so we omit further details.

Finally, we observe that the probability there is no giant component is $O(n^{-2})$, so we can convert to the original model as in Theorem 3. □

References

[AS00] Noga Alon and Joel H. Spencer, *The probabilistic method*, second ed., Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley-Interscience [John Wiley & Sons], New York, 2000, With an appendix on the life and work of Paul Erdős. MR 2003f:60003

[BF01] Tom Bohman and Alan Frieze, *Avoiding a giant component*, Random Structures Algorithms **19** (2001), no. 1, 75–85. MR 2002g:05169

[BFW02] Tom Bohman, Alan Frieze, and Nicholas C. Wormald, *Avoiding a giant component II*, manuscript, 2002.

[BK03] Tom Bohman and David Kravitz, personal communication, 2003.

[Fei98] Uriel Feige, *A threshold of* $\ln n$ *for approximating set cover*, J. ACM **45** (1998), no. 4, 634–652. MR 2000f:68049

[Hoe63] Wassily Hoeffding, *Probability inequalities for sums of bounded random variables*, J. Amer. Statist. Assoc. **58** (1963), 13–30. MR 26 #1908

[JŁR00] Svante Janson, Tomasz Łuczak, and Andrzej Rucinski, *Random graphs*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley-Interscience, New York, 2000. MR 2001k:05180

[McD89] Colin McDiarmid, *On the method of bounded differences*, Surveys in combinatorics, 1989 (Norwich, 1989), London Math. Soc. Lecture Note Ser., vol. 141, Cambridge Univ. Press, Cambridge, 1989, pp. 148–188. MR 91e:05077

[Pit90] Boris Pittel, *On tree census and the giant component in sparse random graphs*, Random Structures Algorithms **1** (1990), no. 3, 311–342. MR 92f:05087

[SSS02] Mark Scharbrodt, Thomas Schickinger, and Angelika Steger, *A new average case analysis for completion time scheduling*, Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC), 2002, pp. 170–178.