

TRIANGLE-SATURATED GRAPHS IN THE SEMI-RANDOM GRAPH PROCESS

FELIX CHRISTIAN CLEMEN AND PAWEŁ PRALAT

ABSTRACT. The semi-random graph process is an adaptive random graph process in which an online algorithm is initially given an empty graph on n vertices. In each round, a vertex u is presented to the algorithm independently and uniformly at random. The algorithm then adaptively selects a vertex v , and adds the edge uv to the graph. We also consider the offline version of the process in which the algorithm is given the entire sequence of random vertex choices before the selection takes place. For a given graph property, the objective of the algorithm is to force the graph to satisfy this property asymptotically almost surely in as few rounds as possible.

In this paper, we focus on the property of being triangle-saturated and establish upper and lower bounds on the number of rounds required to construct a triangle-saturated graph in both the online and offline versions of the process.

1. INTRODUCTION AND MAIN RESULTS

The *semi-random graph process* was suggested by Peleg Michaeli, formally introduced in 2020 [5], and studied recently [2, 3, 4, 10, 20, 23, 25, 27], especially in the context of Hamiltonian cycles [15, 16, 18, 19, 27]. It is an example of an *adaptive* random graph process, in that an *algorithm* has partial control over which random edges are added in each step. Specifically, the algorithm begins with the empty multigraph H_0 on vertex set $[n] = \{1, \dots, n\}$, and in each *step* (or round) $t \in \mathbb{N}$, a vertex u_t is chosen independently and uniformly at random (u.a.r.) from $[n]$. The algorithm is *online*, in that it is given the graph H_{t-1} and vertex u_t and must select a vertex $v_t \in [n]$ and add the edge $u_t v_t$ to H_{t-1} to form H_t . Thus, it decides on v_t without full knowledge of which vertices will be randomly drawn in the future.

In this paper, the goal of the online algorithm is to make the underlying simple graph G_t satisfy a given graph property $\mathcal{P}$ as quickly as possible. Here, G_t is obtained from H_t by deleting loops and replacing each collection of parallel edges by a single edge. If the online algorithm chooses v_t u.a.r. from $[n] \setminus \{u_t\}$ in every round, then each edge of K_n is selected independently and uniformly at random. Thus, this strategy produces the Erdős–Rényi random graph process with replacement. The main focus in the literature is to understand how intelligent decision-making can accelerate the appearance of graph properties $\mathcal{P}$.

There are some interesting variants of the semi-random process, including the *offline* version that was introduced in [5] together with the original model. In this variant of the process, the algorithm is given the entire sequence of random vertex choices before the selection takes place. Clearly, this variant is more powerful and typically one can achieve property $\mathcal{P}$ faster compared to the original model. However, there are properties for which this additional power might not accelerate the process. For example, both online and offline models need $n - 1$ rounds to create a connected graph.

While the semi-random graph process has been studied extensively in recent years, discrete processes in which an algorithm has partial control over its random steps have been studied previously. One of the first such examples is the work of Azar et al. [1] in the context of the balls-into-bins problem. By allowing an algorithm a small amount of *adaptivity*, Azar et al. proved that the maximum

load on any bin can be reduced by an exponential factor in comparison to a purely random (and non-adaptive) strategy. This phenomenon has since been described as the “power of two choices”.

The *Achlioptas process* is another example of an adaptive random process; it was proposed by Dimitris Achlioptas and first formally studied in [6]. The Achlioptas process, too, begins with the empty graph G_0 on vertex set $[n]$. In each round $t \in \mathbb{N}$, the online algorithm is presented two distinct edges e_t^1, e_t^2 drawn u.a.r. from the edges on vertex set $[n]$ that were *not* previously chosen (i.e., not in G_{t-1}). The online algorithm then chooses precisely one of e_t^1, e_t^2 , and then adds it to G_{t-1} , yielding the graph G_t . In contrast to the semi-random process, the objective first considered in [6] is to *delay* the construction of a graph satisfying a property $\mathcal{P}$ for as many rounds as possible. Achlioptas asked what can be done if $\mathcal{P}$ corresponds to the existence of a giant component (i.e., a connected component of size $\Omega(n)$). Bohman and Frieze analyzed a greedy strategy which does *not* build a giant component for $0.535n$ rounds, which is strictly larger than the threshold at which a giant component appears in the Erdős–Rényi random graph process [13]. Improvements have since been made on determining the optimal algorithm for delaying the appearance of a giant component. The best known lower bound of $0.829n$ is due to Spencer and Wormald [29], and the best known upper bound of $0.944n$ is due to Cobărzan [11]. Numerous works have studied other properties and extensions of the Achlioptas process. Most related to our work is [24], whose goal is to force G_t to be Hamiltonian as *quickly* as possible. We refer the reader to the introduction of [14] for an in-depth overview of the literature on the Achlioptas process.

1.1. Main Result: Creating Triangle-saturated Graphs. In this paper, we focus on the property of being triangle-saturated. A *triangle-saturated* (or K_3 -saturated) graph is a graph that is triangle-free but adding any missing edge creates at least one new triangle. This class of graphs is also widely known as maximally triangle-free.

Classical extremal results for triangle-saturated graphs provide useful benchmarks for determining how quickly such graphs can be constructed in the semi-random process. The maximum number of edges in a triangle-saturated graph on n vertices is $\lfloor n^2/4 \rfloor$, and the unique extremal graph is the complete bipartite graph $K_{\lfloor n/2 \rfloor, \lfloor n/2 \rfloor}$ by Mantel’s Theorem [26]. On the other hand, the minimum number of edges in a triangle-saturated graph on n vertices is $n-1$, which immediately gives a lower bound of $n-1$ rounds for the semi-random process. The unique graph attaining this minimum is the star $K_{1, n-1}$ by the Erdős–Hajnal–Moon Theorem [12].

To create a triangle-saturated graph in the semi-random process, it is tempting to simulate the famous triangle-free process that was introduced by Bollobás and Erdős (see [9]). The process starts with an empty graph on n vertices and adds edges one-by-one, chosen uniformly at random, provided that the newly added edge does not form a triangle. The process terminates when no more edges can be added (resulting in a random triangle-saturated graph). Bohman and Keevash proved that *w.h.p.*¹ the triangle-free process terminates after $(1/(2\sqrt{2}) + o(1))(\log n)^{1/2}n^{3/2}$ rounds [8].

To simulate the triangle-free process, in each round the online semi-random process may choose a vertex r u.a.r. from $[n]$ and let $v_t = r$ if $r \neq u_t$, $u_t r \notin E(G_{t-1})$, and adding $u_t r$ does not create a triangle; otherwise, let $v_t = u_t$ (that is, create a loop). Clearly, this coupling ensures that after t rounds, the graph obtained by the semi-random process after deleting the loops is exactly the graph obtained by the triangle-free process after $t-s$ rounds, where s denotes the number of loops created during the first t rounds. Initially, very few loops are created but later on the semi-random process will be mostly creating loops. Hence, this approach will not give any useful upper bound. However, not surprisingly, a fully adaptive algorithm can do much better than that anyway.

Let $w_{\text{on}}(n, H)$ denote the smallest integer m for which there exists an online strategy that constructs an H -saturated graph after m rounds with probability at least $1/2$. Define $w_{\text{off}}(n, H)$ analogously, with the strategy allowed to know the entire sequence of presented vertices in advance.

¹A property is said to hold *with high probability* (*w.h.p.*) if it holds with probability tending to one as $n \rightarrow \infty$

In this paper, we prove the following results that are asymptotic by nature. For example, to prove an upper bound for $w_{\text{on}}(n, H)$ or $w_{\text{off}}(n, H)$, we fix a strategy and a function $u = u(n)$ and show that *w.h.p.* this specific strategy creates an H -saturated graph after u rounds. Similarly, to prove a lower bound, we fix a function $\ell = \ell(n)$ and show that *w.h.p.* no strategy creates an H -saturated graph after ℓ rounds. As a result, the bounds claimed below hold assuming that n is large enough.

For the offline variant, we prove the following bounds.

Theorem 1.1.

$$\begin{aligned} w_{\text{off}}(n, K_3) &\leq n \log n - n \log \log n + n \\ &= n \log n - (1 + o(1))n \log \log n. \end{aligned}$$

Theorem 1.2.

$$\begin{aligned} w_{\text{off}}(n, K_3) &\geq n \log n - 2n \log \log n - n \log 7 \\ &= n \log n - (2 + o(1))n \log \log n. \end{aligned}$$

For the original (online) process, we prove the following bounds.

Theorem 1.3. *Let $\omega = \omega(n)$ be any function tending to infinity as $n \rightarrow \infty$. Then,*

$$w_{\text{on}}(n, K_3) \leq n \log n + n\omega.$$

Theorem 1.4.

$$\begin{aligned} w_{\text{on}}(n, K_3) &\geq n \log n - n \log \log n - 3n \log \log \log n \\ &= n \log n - (1 + o(1))n \log \log n. \end{aligned}$$

Choosing any function $\omega \rightarrow \infty$ such that $\omega = o(\log \log n)$ in Theorem 1.3, we see that in both the online and offline settings the difference between our upper and lower bounds is $(1 + o(1))n \log \log n$. The proofs of the upper bounds reveal a natural bottleneck, suggesting that the upper bounds are closer to the truth, although we were unable to close the remaining gap.

Let us also mention that one should be able to generalize the arguments and extend our results to K_k -saturated graphs for some fixed integer $k \geq 3$. For simplicity, we concentrate on triangle-saturated graphs.

Proofs of Theorems 1.1 and 1.2 can be found in Section 3 whereas proofs of Theorems 1.3 and 1.4 are provided in Section 4.

1.2. Further Related Work. The seminal paper [5] showed that the semi-random graph process is general enough to simulate several well-studied random graph models by using appropriate strategies. In the same paper, the process was studied for various natural properties such as having minimum degree $k \in \mathbb{N}$ or having a fixed graph H as a subgraph. In particular, it was shown that *w.h.p.* one can construct H in less than $n^{(d-1)/d}\omega$ rounds where $d \geq 2$ is the degeneracy of H and $\omega = \omega(n)$ is any function that tends to infinity as $n \rightarrow \infty$. This property was recently revisited in [2], where a conjecture from [5] was proven for any graph H : *w.h.p.* it takes at least $n^{(d-1)/d}/\omega$ rounds to create H . The property of having cliques of order tending to infinity as $n \rightarrow \infty$ was investigated in [17]. In [23], k -factors and k -connectivity were studied.

Another property studied in the context of semi-random processes is that of having a perfect matching. Since the 2-out process has a perfect matching *w.h.p.* [30], and the semi-random process can simulate the 2-out process, we immediately get an upper bound of $(2 + o(1))n$. By simulating the semi-random process with another random graph process known to have a perfect matching *w.h.p.* [22], the bound can be improved to $(1 + 2/e + o(1))n < 1.73576n$ [5]. This bound was improved by investigating another fully adaptive algorithm [20], giving the current best bound of

1.20524 n . The same paper improves the lower bound observed in [5] of $(\ln(2) + o(1))n > 0.69314n$ to 0.93261 n .

Next, we discuss results for the property of having a Hamiltonian cycle. It was shown in [7] that the 3-out process generates a random graph that is Hamiltonian *w.h.p.* Since the semi-random process can simulate the 3-out process, we get an upper bound of $(3 + o(1))n$. A new upper bound was obtained in [18] in terms of an optimal solution to an optimization problem whose value is believed to be at most 2.61135 n by numerical support. The upper bound $(3 + o(1))n$ obtained by simulating the 3-out process is *non-adaptive*. That is, the strategy does not depend on the history of the semi-random process. The improvement in [18] is adaptive but in a weak sense. The strategy consists of four phases, each lasting a linear number of rounds, and the strategy is adjusted only at the end of each phase: for example, the algorithm might identify vertices of low degree, and then focus on them during the next phase. In [19], a fully adaptive strategy was proposed: at every step t , it pays attention to G_{t-1} and u_t . As expected, such a strategy creates a Hamiltonian cycle substantially faster, and it improves the upper bound from 2.61135 n to 2.01678 n . A further improvement in [16] brings the upper bound down to 1.84887 n . All ideas are combined together in [15] to reduce it further, to 1.81701 n .

For the lower bound, it was observed in [5] that because any Hamiltonian graph has minimum degree at least 2, the lower bound is at least $(\ln 2 + \ln(1 + \ln 2) + o(1))n \geq 1.21973n$. This was subsequently improved by $10^{-8}n$ in [18]. By identifying structures created by the semi-random process whose edges cannot all lie on a Hamiltonian cycle, the bound was further raised to 1.26575 n in [15, 19].

Other adaptive random graph processes and variants of the semi-random graph process have been considered in the literature. The *semi-random tree process* is introduced in [10], where in each round, a random spanning tree of K_n is presented to the algorithm, who chooses one of the edges to keep. In [28], k random vertices rather than just one are offered, and the algorithm chooses one of them before creating an edge. In [25], a general definition of an *adaptive random graph process* is proposed. By parameterizing it appropriately, one recovers the Achlioptas process, the semi-random graph process, as well as the models of [10] and [28]. In [21], the vertices offered by the process follow a random permutation. Finally, hypergraphs are investigated in [2, 3, 27].

2. NOTATION AND AUXILIARY OBSERVATIONS

Throughout the paper, all logarithms are natural. Whenever a non-integer expression is used to specify a number of rounds, we take its floor.

2.1. Notation. An unordered pair $ab \notin E(G)$ in a triangle-free graph G is called *open* if adding ab to G does not create a triangle. Note that a triangle-free graph is triangle-saturated if and only if it does not contain an open pair. An unordered pair $ab \notin E(G)$ in a triangle-free graph G is called *closed* if adding ab to G creates a triangle.

For a graph G and a vertex $x \in V(G)$, we write $N_G(x) := \{y \in V(G) : xy \in E(G)\}$ for the *neighbourhood* of x in G , and $\deg_G(x) := |N_G(x)|$ for its *degree*.

2.2. Coupon Collector Estimates. The number of vertices presented by the semi-random process is well-understood as it is related to the famous coupon collector problem. We record a few properties that we will use in our proofs.

Lemma 2.1. *Fix any function $T = T(n)$. For any vertex $i \in [n]$, let X_i be the random variable counting how many times vertex i is presented by the semi-random process during the first T steps. Let Z be the random variable counting how many vertices were not presented by the semi-random process during the first T steps. Then, the following properties hold.*

(a) Suppose that $T/n - \log n \rightarrow -\infty$ as $n \rightarrow \infty$. Then, *w.h.p.*

$$Z = (1 + o(1))n \exp\left(-\frac{T}{n}\right) \rightarrow \infty.$$

(b) Suppose that $T/n - \log n \rightarrow \infty$ as $n \rightarrow \infty$. Then, *w.h.p.* $Z = 0$.

(c) Suppose that $T = (1 + o(1))n \log n$. Then, *w.h.p.* no vertex is presented more than $3 \log n$ times, that is,

$$\max_{i \in [n]} X_i \leq 3 \log n.$$

Proof. We start with the proof of part (a). Suppose that $T = T(n) = n(\log n - \omega)$, where $\omega = \omega(n) \rightarrow \infty$ as $n \rightarrow \infty$. Clearly, for any $i \in [n]$, the probability that vertex i is *not* presented by the semi-random process during the first T steps (that is, that $u_t \neq i$ for all $t \in [T]$) can be estimated as follows:

$$\mathbb{P}(X_i = 0) = \left(1 - \frac{1}{n}\right)^T = \exp\left(-\frac{T}{n} + O\left(\frac{T}{n^2}\right)\right) = (1 + o(1)) \exp\left(-\frac{T}{n}\right),$$

and so $\mathbb{E}[Z] = (1 + o(1))n \exp(-T/n) = (1 + o(1))e^\omega \rightarrow \infty$ as $n \rightarrow \infty$. Similarly,

$$\mathbb{E}[Z(Z-1)] = n(n-1) \left(1 - \frac{2}{n}\right)^T = (1 + o(1))n^2 \exp\left(-\frac{2T}{n}\right) = (1 + o(1))(\mathbb{E}[Z])^2,$$

and so $\text{Var}[Z] = o(\mathbb{E}[Z])^2$. It follows from Chebyshev's inequality that *w.h.p.*

$$Z = (1 + o(1))\mathbb{E}[Z] = (1 + o(1))n \exp(-T/n).$$

This finishes the proof of part (a).

For part (b), suppose that $T = T(n) = n(\log n + \omega)$, where $\omega = \omega(n) \rightarrow \infty$ as $n \rightarrow \infty$. Arguing as before and using $1 - x \leq e^{-x}$, we obtain

$$\mathbb{E}[Z] = n \left(1 - \frac{1}{n}\right)^T \leq n \exp\left(-\frac{T}{n}\right) = \exp\left(\log n - \frac{T}{n}\right) \rightarrow 0.$$

Hence, by Markov's inequality, we get that *w.h.p.* $Z = 0$. This finishes the proof of part (b).

Finally, for part (c), suppose that $T = (1 + o(1))n \log n$ and fix any vertex $i \in [n]$. Clearly, $\mathbb{E}[X_i] = T/n = (1 + o(1)) \log n$. It follows from Chernoff's bound that

$$\begin{aligned} \mathbb{P}(X_i \geq 3 \log n) &= \mathbb{P}\left(X_i \geq \mathbb{E}[X_i] + (2 + o(1)) \log n\right) \\ &\leq \exp\left(-\left(\frac{2^2}{2(1 + 2/3)} + o(1)\right) \log n\right) = o(1/n). \end{aligned}$$

Hence, by the union bound over n vertices, *w.h.p.* $X_i \leq 3 \log n$ for all $i \in [n]$. This finishes the proof of part (c). $\square$

3. OFFLINE VARIANT – PROOF OF THEOREMS 1.1 AND 1.2

Let us start with the upper bound as it is simpler.

Proof of Theorem 1.1. Fix $T = n(\log n - \log \log n + 1) = (1 + o(1))n \log n$. Let Z_T be the number of vertices not presented during the first T steps. It follows from Lemma 2.1(a) that *w.h.p.* $Z_T = (1 + o(1))n \exp(-T/n) = (1 + o(1))e^{-1} \log n$.

Now, let Y_T be the random variable counting how many times vertex 1 is presented by the semi-random process during the first T steps. Clearly, $Y_T \sim \text{Bin}(T, 1/n)$ with $\mathbb{E}[Y_T] = T/n = (1 + o(1)) \log n$. It follows immediately from Chernoff's inequality that *w.h.p.* $Y_T = (1 + o(1))\mathbb{E}[Y_T] = (1 + o(1)) \log n$.

Our strategy is simple. We build a star centred at vertex 1, which is a triangle-saturated graph. Each time vertex 1 is presented by the semi-random process, we connect it to one of the vertices that are not presented during the first T steps. On the other hand, each time a vertex other than 1 is presented, we connect it with 1. Since *w.h.p.* vertex 1 is presented more often than the number of vertices not presented at all (*w.h.p.* $(1 + o(1)) \log n = Y_T > Z_T = (1 + o(1))e^{-1} \log n$), the strategy works *w.h.p.* This completes the proof of the theorem. $\square$

Remark 1. Note that in the proof of the above theorem we could have selected a vertex presented the most often by the semi-random process instead of vertex 1. This vertex could take care of slightly more vertices that were not presented at all which, in turn, would improve our upper bound. However, *w.h.p.* each vertex $i \in [n]$ is presented $O(\log n)$ times (see Lemma 2.1(c)) so this would only improve the bound by an additive term $O(n)$. Hence, we opted for a slightly easier proof.

We now move to the lower bound.

Proof of Theorem 1.2. Fix $T = n(\log n - 2 \log \log n - \log 7) = (1 + o(1))n \log n$. Let G_T be the graph constructed by an arbitrary offline strategy. If G_T contains a triangle, then it is not triangle-saturated, and there is nothing to prove. We may therefore assume that G_T is triangle-free.

As in the proof of Theorem 1.1, let Z_T be the number of vertices *not* presented by the semi-random process during the first T steps. It follows from Lemma 2.1(a) that *w.h.p.* $Z_T = (7 + o(1)) \log^2 n$. Let X_i be the random variable counting how many times vertex $i \in [n]$ is presented by the semi-random process during the first T steps. It follows from Lemma 2.1(c) that *w.h.p.* $\max_{i \in [n]} X_i \leq 3 \log n$. Since we aim for a statement that holds *w.h.p.*, we may assume that these two properties hold.

Let $I \subseteq [n]$ be the set of vertices not presented by the semi-random process. Clearly, I induces an independent set in G_T and, by our assumption, $|I| = Z_T = (7 + o(1)) \log^2 n$. Our goal is to show that there exists an open pair ab , where $a \in I^c$ and $b \in I$. This will complete the proof of the theorem, since such a pair witnesses that G_T is not triangle-saturated.

Suppose that a cross-pair ab , with $a \in I^c$ and $b \in I$, is closed. Then, there exists a vertex $w \in [n]$ such that $wa, wb \in E(G_T)$. Since I is an independent set, $w \in I^c$. Therefore, the number of closed cross-pairs ab is at most

$$\sum_{w \in I^c} |N_{G_T}(w) \cap I^c| \cdot |N_{G_T}(w) \cap I|. \quad (1)$$

Note that a closed pair may be counted multiple times (if multiple common neighbours exist) in (1), but this only strengthens the upper bound. Every edge between w and I must be created in a round in which w is presented. Hence, for every $w \in I^c$,

$$|N_{G_T}(w) \cap I| \leq X_w \leq \max_{i \in [n]} X_i \leq 3 \log n,$$

where the last inequality follows from our assumption. Consequently, the number of closed cross-pairs ab is at most

$$\begin{aligned} \sum_{w \in I^c} |N_{G_T}(w) \cap I^c| \cdot |N_{G_T}(w) \cap I| &\leq 3 \log n \cdot \sum_{w \in I^c} |N_{G_T}(w) \cap I^c| \leq 3 \log n \cdot \sum_{w \in I^c} \deg_{G_T}(w) \\ &\leq 3 \log n \cdot 2T \leq (6 + o(1))n \log^2 n. \end{aligned}$$

Moreover, trivially, the number of pairs ab such that $ab \in E(G_T)$ is at most T . Hence the number of cross-pairs that are either edges or closed is at most

$$T + (6 + o(1))n \log^2 n = (6 + o(1))n \log^2 n < (7 + o(1))n \log^2 n = |I||I^c|.$$

It follows that some pair between I and I^c is open (neither an edge nor closed). Hence, G_T is not triangle-saturated. This completes the proof of the theorem. $\square$

4. ONLINE VARIANT – PROOF OF THEOREMS 1.3 AND 1.4

As in the previous section, we start with the upper bound.

Proof of Theorem 1.3. Let $\omega = \omega(n)$ be any function tending to infinity as $n \rightarrow \infty$. Fix $T = n(\log n + \omega)$. It follows from Lemma 2.1(b) that *w.h.p.* each vertex is presented at least once.

We can build a star very easily, creating a triangle-saturated graph. We can pick an arbitrary vertex (say, vertex n) before the process starts. Then, in each step of the process we connect the presented vertex u_t to this chosen vertex (that is, fix $v_t = n$ for all $t \in [T]$). The desired star is created in at most T steps *w.h.p.*, which finishes the proof of the theorem. $\square$

Remark 2. Note that in the proof of the above theorem, one could consider the following, slightly better strategy. One may still connect vertices presented by the semi-random process to vertex n , unless n is presented. If n is presented, then one may connect it to an arbitrary vertex that is not yet connected to n . This would speed up the creation of a star but not by much. The original strategy is expected to finish in $\sum_{i=1}^{n-1} \frac{n}{i} = nH_n + O(1) = n \log n + \gamma n + O(1)$ steps, where H_n is the harmonic number and $\gamma \approx 0.57721$ is the Euler-Mascheroni constant. The better strategy is expected to finish in $\sum_{i=1}^{n-1} \frac{n}{i+1} = nH_n - n = n \log n - (1 - \gamma)n + O(1)$ steps, improving the expectation only by a linear term.

We now move to the lower bound which is much more involved.

Proof of Theorem 1.4. We will analyze the process in two phases. Let us fix

$$T_1 = n(\log n - 3 \log \log n) \quad \text{and} \quad T = n(\log n - \log \log n - 3 \log \log \log n).$$

Fix an arbitrary online strategy, and let G_t denote the graph constructed after t rounds. If G_T contains a triangle, then it is not triangle-saturated, and there is nothing to prove. We may therefore assume that G_T is triangle-free.

Let Z_{T_1} be the number of vertices *not* presented by the semi-random process during the first T_1 steps. It follows from Lemma 2.1(a) that *w.h.p.* $Z_{T_1} = (1 + o(1)) \log^3 n$. For $s \in [T]$ and $v \in [n]$, let

$$M_s(v) := |\{t \in [s] : u_t = v\}|$$

be the random variable counting how many times vertex v is presented by the semi-random process during the first s rounds. It follows from Lemma 2.1(c) that *w.h.p.* $\max_{v \in [n]} M_T(v) \leq 3 \log n$.

Let $I := \{v \in [n] : M_{T_1}(v) = 0\}$ be the set of vertices not presented by the semi-random process during the first T_1 steps. Since *w.h.p.* $|I| = Z_{T_1} = (1 + o(1)) \log^3 n$, *w.h.p.* there are $(1 + o(1))n \log^3 n$ cross pairs ab with $a \in I$ and $b \in I^c$. Arguing exactly as in the proof of Theorem 1.2, regardless of how graph G_{T_1} is constructed, we conclude that *w.h.p.* almost all of these pairs are still open.

Claim 4.1. *The following holds w.h.p.: the number of cross pairs ab , $a \in I, b \in I^c$ that are edges or closed at time T_1 is $O(n \log^2 n)$.*

Proof. The proof is identical to the corresponding argument in the proof of Theorem 1.2, with T replaced by T_1 . $\square$

The above claim shows that on average a vertex from I is part of $O(n/\log n)$ cross pairs that are edges or closed. Clearly, there could be some vertices in I that are part of, say, more than $n/\log \log n$ such cross pairs. However, a simple averaging argument shows that there cannot be too many such vertices. Formally, for each $a \in I$, let

$$c(a) := |\{b \in I^c : ab \text{ is an edge or is closed in } G_{T_1}\}|,$$

and define

$$I' := \left\{ a \in I : c(a) \leq \frac{n}{\log \log n} \right\}.$$

Claim 4.2. *The following holds w.h.p.: $|I'| = (1 + o(1))|I|$.*

Proof. By Claim 4.1, w.h.p. $\sum_{a \in I} c(a) = O(n \log^2 n)$. On the other hand, by the definition of I' ,

$$\sum_{a \in I \setminus I'} c(a) > |I \setminus I'| \frac{n}{\log \log n}.$$

Consequently, w.h.p. $|I \setminus I'| = O((\log n)^2 (\log \log n)) = o(\log^3 n)$. Since $|I| = (1 + o(1)) \log^3 n$ w.h.p., we conclude that w.h.p. $|I'| = (1 + o(1))|I|$. $\square$

Let $J \subseteq [n]$ be the set of vertices not presented by the semi-random process by time T , that is, $J := \{v \in [n] : M_T(v) = 0\}$. It follows from Lemma 2.1(a) that w.h.p.

$$|J| = Z_T = (1 + o(1))(\log n)(\log \log n)^3.$$

Note that $J \subseteq I$ because $T \geq T_1$. More importantly, J is a random subset of I of cardinality Z_T . Consider $J' = J \cap I'$. By Claim 4.2, almost all vertices in I are in I' .

Conditioned on the history up to time T_1 and on $|J|$, the set J is uniformly distributed among all $|J|$ -subsets of I . Consequently,

$$\mathbb{E}[|J \setminus I'| \mid I, I', |J|] = |J| \frac{|I \setminus I'|}{|I|} = o(|J|),$$

where the last equality follows from Claim 4.2. Thus, by Markov's inequality, $|J \setminus I'| = o(|J|)$ w.h.p. Therefore,

$$|J'| = (1 + o(1))|J| = (1 + o(1))(\log n)(\log \log n)^3.$$

Next, we bound the number of cross pairs between J' and J'^c that are edges or closed.

Claim 4.3. *The following holds w.h.p.: at time T_1 , the number of pairs between J' and J'^c that are edges or closed is $o(n|J'|)$.*

Proof. Since $J' \subseteq I'$, there are at most

$$\sum_{a \in J'} c(a) \leq \frac{n|J'|}{\log \log n} = o(n|J'|)$$

such pairs between J' and I^c . The remaining pairs lie between J' and $I \setminus J'$, of which there are at most $|J'| |I| = O(|J'| \log^3 n) = o(n|J'|)$. $\square$

We now upper bound the number of additional cross pairs between J' and J'^c that become closed during rounds $T_1 + 1, T_1 + 2, \dots, T$.

Lemma 4.4. *The following holds w.h.p.: the number of cross pairs between J' and J'^c that become closed during rounds $T_1 + 1, T_1 + 2, \dots, T$ is $O(n(\log n)(\log \log n)^2)$.*

Proof. For a set $S \subseteq [n]$, write

$$d_t(x, S) := |N_{G_t}(x) \cap S| \quad \text{and} \quad d_t(x) := d_t(x, [n]) = \deg_{G_t}(x).$$

Suppose that the edge $u_t v_t$ is added at time t . Every pair that becomes closed at time t is either of the form $x v_t$, where $x \in N_{G_{t-1}}(u_t)$, or of the form $x u_t$, where $x \in N_{G_{t-1}}(v_t)$. Since $u_t \notin J'$, the newly closed cross pairs of the second type are counted by $d_{t-1}(v_t, J')$. For the first type, if $v_t \in J'$, there are at most $d_{t-1}(u_t)$ such pairs, whereas if $v_t \notin J'$, there are at most $d_{t-1}(u_t, J')$ such pairs. Thus, the number of cross pairs between J' and J'^c that become closed at time t is at most

$$\mathbf{1}_{\{v_t \in J'\}} d_{t-1}(u_t) + d_{t-1}(u_t, J') + d_{t-1}(v_t, J').$$

Consequently, the total number C_{new} of such newly closed cross pairs satisfies the following:

$$C_{\text{new}} \leq \sum_{t=T_1+1}^T \mathbf{1}_{\{v_t \in J'\}} d_{t-1}(u_t) + \sum_{t=T_1+1}^T d_{t-1}(u_t, J') + \sum_{t=T_1+1}^T d_{t-1}(v_t, J').$$

Because no vertex of J' is ever presented by the semi-random process, every edge between a vertex x and J' must be created in a round in which x is presented. Hence, *w.h.p.* for every $s \leq T$ and for any $x \in [n]$, $d_s(x, J') \leq M_s(x) \leq M_T(x) \leq 3 \log n$. It follows that *w.h.p.*

$$\begin{aligned} C_{\text{new}} &\leq \sum_{t=T_1+1}^T \mathbf{1}_{\{v_t \in J'\}} d_{t-1}(u_t) + (T - T_1)6 \log n \\ &\leq \sum_{t=T_1+1}^T \mathbf{1}_{\{v_t \in J'\}} d_{t-1}(u_t) + 12n(\log n)(\log \log n). \end{aligned} \quad (2)$$

It remains to estimate the sum in (2). Let

$$A := \sum_{t=T_1+1}^T \mathbf{1}_{\{v_t \in J'\}} d_{t-1}(u_t) \leq \sum_{t=T_1+1}^T d_{t-1}(u_t).$$

Therefore,

$$\mathbb{E}A \leq \sum_{t=T_1+1}^T \mathbb{E}[d_{t-1}(u_t)].$$

Since u_t is a vertex selected uniformly at random from $[n]$ and independent of G_{t-1} ,

$$\mathbb{E}[d_{t-1}(u_t) \mid G_{t-1}] = \frac{2e(G_{t-1})}{n} \leq \frac{2(t-1)}{n} \leq 2 \log n.$$

Hence,

$$\mathbb{E}A \leq (2 \log n)(T - T_1) \leq 4n(\log n)(\log \log n).$$

By Markov's inequality, *w.h.p.* $A = O(n(\log n)(\log \log n)^2)$. Combining this with (2), we conclude that *w.h.p.* $C_{\text{new}} = O(n(\log n)(\log \log n)^2)$. $\square$

With Lemma 4.4 at hand, it is very easy to complete the proof of Theorem 1.4. By Lemma 4.4 and Claim 4.3, at time T , *w.h.p.* the total number of cross pairs between J' and J'^c that are closed is $o(n|J'|) + O(n(\log n)(\log \log n)^2) = o(n|J'|)$. The number of edges between J' and J'^c is, trivially, at most $T = O(n \log n) = o(n|J'|)$. On the other hand, *w.h.p.* the total number of cross pairs is equal to $|J'| |J'^c| = (1 + o(1))n|J'|$. It follows that *w.h.p.* some cross pair between J' and J'^c is neither an edge nor closed. Hence G_T is not triangle-saturated *w.h.p.*, completing the proof. $\square$

5. ACKNOWLEDGEMENTS

The first author's research is supported by a PIMS Postdoctoral Fellowship (PIMS-20260806-PDF).

REFERENCES

- [1] Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. Balanced allocations. *SIAM Journal on Computing*, 29(1):180–200, 1999.
- [2] Natalie Behague, Trent Marbach, Paweł Prałat, and Andrzej Ruciński. Subgraph games in the semi-random graph process and its generalization to hypergraphs. *Electronic Journal of Combinatorics*, 31(3):#P3.30, 2024.
- [3] Natalie Behague, Paweł Prałat, and Andrzej Ruciński. Creating subgraphs in semi-random hypergraph games. *The Electronic Journal of Combinatorics*, pages P1–22, 2026.
- [4] Omri Ben-Eliezer, Lior Gishboliner, Dan Hefetz, and Michael Krivelevich. Very fast construction of bounded-degree spanning graphs via the semi-random graph process. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5–8, 2020*, pages 718–737. SIAM, 2020.
- [5] Omri Ben-Eliezer, Dan Hefetz, Gal Kronenberg, Olaf Parczyk, Clara Shikhelman, and Miloš Stojaković. Semi-random graph process. *Random Structures & Algorithms*, 56(3):648–675, 2020.
- [6] Tom Bohman and Alan Frieze. Avoiding a giant component. *Random Structures & Algorithms*, 19(1):75–85, 2001.

- [7] Tom Bohman and Alan Frieze. Hamilton cycles in 3-out. *Random Structures & Algorithms*, 35(4):393–417, 2009.
- [8] Tom Bohman and Peter Keevash. Dynamic concentration of the triangle-free process. *Random Structures & Algorithms*, 58(2):221–293, 2021.
- [9] Béla Bollobás and Oliver Riordan. Random graphs and branching processes. In *Handbook of large-scale random networks*, pages 15–115. Springer, 2009.
- [10] Sofiya Burova and Lyuben Lichev. The semi-random tree process. *European Journal of Combinatorics*, 126:104120, 2025.
- [11] Petruș Cobârzan. A new upper bound for Achlioptas processes. *arXiv preprint arXiv:1410.3876*, 2014.
- [12] Paul Erdős, András Hajnal, and John W. Moon. A problem in graph theory. *The American Mathematical Monthly*, 71(10):1107–1110, 1964.
- [13] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5(1):17–61, 1960.
- [14] Nicolas Fraiman, Lyuben Lichev, and Dieter Mitsche. On the power of choice for boolean functions. *SIAM Journal on Discrete Mathematics*, 36(3):1943–1954, 2022.
- [15] Alan Frieze, Pu Gao, Calum MacRury, Paweł Pralat, and Gregory B Sorkin. Building Hamiltonian cycles in the semi-random graph process in less than $2n$ rounds. *European Journal of Combinatorics*, 126:104122, 2025.
- [16] Alan Frieze and Gregory B Sorkin. Hamilton cycles in a semi-random graph model. *arXiv preprint arXiv:2208.00255*, 2022.
- [17] David Gamarnik, Mi Hyun Kang, and Paweł Pralat. Cliques, chromatic number, and independent sets in the semi-random process. *SIAM Journal on Discrete Mathematics*, 38(3):2312–2334, 2024.
- [18] Pu Gao, Bogumił Kamiński, Calum MacRury, and Paweł Pralat. Hamilton cycles in the semi-random graph process. *European Journal of Combinatorics*, 99:103423, 2022.
- [19] Pu Gao, Calum MacRury, and Paweł Pralat. A fully adaptive strategy for Hamiltonian cycles in the semi-random graph process. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19–21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference)*, volume 245 of *LIPICs*, pages 29:1–29:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- [20] Pu Gao, Calum MacRury, and Paweł Pralat. Perfect matchings in the semirandom graph process. *SIAM Journal on Discrete Mathematics*, 36(2):1274–1290, 2022.
- [21] Shoni Gilboa and Dan Hefetz. Semi-random process without replacement. In *Extended Abstracts EuroComb 2021*, pages 129–135. Springer, 2021.
- [22] Michał Karoński, Ed Overman, and Boris Pittel. On a perfect matching in a random digraph with average out-degree below two. *Journal of Combinatorial Theory, Series B*, 143, 03 2020.
- [23] Hidde Koerts. k -connectedness and k -factors in the semi-random graph process. Master’s thesis, University of Waterloo, 2022.
- [24] Michael Krivelevich, Eyal Lubetzky, and Benny Sudakov. Hamiltonicity thresholds in Achlioptas processes. *Random Structures & Algorithms*, 37(1):1–24, 2010.
- [25] Calum MacRury and Erlang Surya. Sharp thresholds in adaptive random graph processes. *Random Structures & Algorithms*, 64(3):741–767, 2024.
- [26] Willem Mantel. Vraagstuk 28. *Wiskundige Opgaven*, 10:60–61, 1907.
- [27] Michael Molloy, Paweł Pralat, and Gregory B Sorkin. Perfect matchings and loose Hamilton cycles in the semirandom hypergraph model. *Random Structures & Algorithms*, 66(2):e21270, 2025.
- [28] Paweł Pralat and Harjas Singh. Power of k choices in the semi-random graph process. *Electronic Journal of Combinatorics*, 31(1):#P1.11, 2024.
- [29] Joel Spencer and Nicholas Wormald. Birth control for giants. *Combinatorica*, 27, 07 2004.
- [30] David W. Walkup. Matchings in random regular bipartite digraphs. *Discrete Mathematics*, 31(1):59–64, 1980.

DEPARTMENT OF MATHEMATICS AND STATISTICS, UNIVERSITY OF VICTORIA, VICTORIA, CANADA.
 Email address: fclemen@uvic.ca

DEPARTMENT OF MATHEMATICS, TORONTO METROPOLITAN UNIVERSITY, TORONTO, CANADA
 Email address: pralat@torontomu.ca