BUILDING HAMILTONIAN CYCLES IN THE SEMI-RANDOM GRAPH PROCESS IN LESS THAN $2n$ ROUNDS

ALAN FRIEZE, PU GAO, CALUM MACRURY, PAWEŁ PRAŁAT, AND GREGORY B. SORKIN

ABSTRACT. The semi-random graph process is an adaptive random graph process in which an online algorithm is initially presented 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. 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.

We focus on the property of Hamiltonicity. We present an adaptive strategy which creates a Hamiltonian cycle in αn rounds, where $\alpha < 1.81696$ is derived from the solution to a system of differential equations. We also show that achieving Hamiltonicity requires at least βn rounds, where $\beta > 1.26575$.

1. INTRODUCTION AND MAIN RESULTS

The *semi-random graph process* was suggested by Peleg Michaeli, introduced formally in 2020 [5], and studied recently [4, 19, 2, 11, 24, 22, 25, 3], especially in the context of Hamiltonian cycles [17, 18, 15, 25]. 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 graph G_0 on vertex set $[n] = \{1, \ldots, n\}$, 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 G_{t-1} and vertex u_t and must select a vertex $v_t \in [n] \setminus \{u_t\}$ and add the edge (u_t, v_t) to G_{t-1} to form G_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 build a multigraph satisfying a given graph property $\mathcal{P}$ as quickly as possible. Clearly, if the online algorithm chooses v_t u.a.r. for $m \geq 1$ consecutive rounds, then this is the Erdős–Rényi random graph process with multi-edges. The main focus in the literature is understanding how through intelligent decision-making, the online algorithm can speed up the appearance of certain graph properties $\mathcal{P}$.

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 bin-packing 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 phenomena 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 [9]. 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 [9] 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 [27], and the best known upper bound of $0.944n$ is due to Cobârzan [12]. Numerous works have studied other properties and extensions of the Achlioptas process. Most related to our work is [23], 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. **Definitions.** We now introduce some definitions and notation for the semi-random graph process. We formalize an online algorithm using a *strategy* $\mathcal{S}$. A strategy $\mathcal{S}$ specifies for each $n \geq 1$, a sequence of functions $(s_t)_{t=1}^{\infty}$, where for each $t \in \mathbb{N}$, $s_t(u_1, v_1, \ldots, u_{t-1}, v_{t-1}, u_t)$ is a distribution on $[n] \setminus \{u_t\}$ which depends on the vertex u_t, and the *history* of the process up until step $t-1$ (i.e., $u_1, v_1, \ldots, u_{t-1}, v_{t-1}$). Then, v_t is chosen according to this distribution, and (u_t, v_t) is added to $G_{t-1}^{\mathcal{S}}(n)$, the multigraph constructed by $\mathcal{S}$ after the first $t-1$ steps. If s_t is an atomic distribution, then v_t is determined by $u_1, v_1, \ldots, u_{t-1}, v_{t-1}, u_t$. We denote by $(G_i^{\mathcal{S}}(n))_{i=0}^{t}$ the sequence of random multigraphs obtained by following the strategy $\mathcal{S}$ for t steps, and we shorten $G_t^{\mathcal{S}}(n)$ to G_t or $G_t(n)$ when clear.

Suppose $\mathcal{P}$ is an *increasing* (i.e., *monotone*) graph property. Given a strategy $\mathcal{S}$ and a constant $0 < q < 1$, let $m_{\mathcal{P}}(\mathcal{S}, q, n)$ be the minimum $t \geq 0$ for which $\mathbb{P}[G_t \in \mathcal{P}] \geq q$, where $m_{\mathcal{P}}(\mathcal{S}, q, n) := \infty$ if no such t exists. Define

$$m_{\mathcal{P}}(q, n) = \inf_{\mathcal{S}} m_{\mathcal{P}}(\mathcal{S}, q, n),$$

where the infimum is over all strategies on $[n]$. As $\mathcal{P}$ is increasing, for each $n \geq 1$, if $0 \leq q_1 \leq q_2 \leq 1$, then $m_{\mathcal{P}}(q_1, n) \leq m_{\mathcal{P}}(q_2, n)$. Thus, the function $q \mapsto \limsup_{n \to \infty} m_{\mathcal{P}}(q, n)$ is non-decreasing, so the limit

$$C_{\mathcal{P}} := \lim_{q \to 1^-} \limsup_{n \to \infty} \frac{m_{\mathcal{P}}(q, n)}{n} \tag{1}$$

is guaranteed to exist. The goal is typically to compute upper and lower bounds on $C_{\mathcal{P}}$ for various properties $\mathcal{P}$.

Remark 1. Note that although $C_{\mathcal{P}}$ is well defined for all increasing properties, it only gives useful information if $C_{\mathcal{P}}$ is not equal to 0 or ∞, i.e., if a linear number (in n) of steps is necessary and sufficient to construct some graph in $\mathcal{P}$. If $C_{\mathcal{P}}$ is equal to 0 or ∞ then the property is not linear and the definition (1) should be adapted, scaling $m_{\mathcal{P}}(q, n)$ by an appropriate function of n rather than n itself in the denominator.

1.2. **Main Results: Upper Bound.** In this paper, we focus on the property of having a Hamiltonian cycle, which we denote by HAM. It was shown in [10] that the the well-known 3-out process generates a random graph that is Hamiltonian a.a.s. (asymptotically almost surely, i.e., with probability tending to 1 as $n \to \infty$). The very first paper on the semi-random process, [5], showed that it can simulate the k-out process, from which it follows that $C_{\text{HAM}} \leq 3$. A new upper bound was obtained in [17] in terms of an optimal solution to an optimization problem whose value is believed to be at most 2.61135 by numerical support.

The upper bound $C_{\text{HAM}} \leq 3$ 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 [17] 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 the proceedings version [18] of this paper, 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 to 2.01678. A neat improvement in [15] brings the upper bound down to 1.84887. In this paper, we combine all the ideas together to reduce it further, to 1.81701.

Theorem 1.1. $C_{HAM} \leq \alpha \leq 1.81701$, where α is derived from a system of differential equations.

The numerical results presented in this paper were obtained using the Julia programming language [8]. We would like to thank Bogumił Kamiński from SGH Warsaw School of Economics for helping us to implement it. The program is available on-line.[1]

1.3. **Main Results: Lower Bound.** Let us now turn to the lower bound. As observed in [5], if G_t has a Hamiltonian cycle, then G_t has minimum degree at least 2; thus $C_{HAM} \geq C_{\mathcal{P}} = \ln 2 + \ln(1 + \ln 2) \geq 1.21973$, where $\mathcal{P}$ is the property of having minimum degree 2. In [17], this was shown to not be tight: it was increased by a numerically negligible 10^{-8}. By investigating some specific structures generated by the semi-random process, containing many edges that cannot simultaneously belong to a Hamiltonian cycle, we improve the lower bound of $\ln 2 + \ln(1 + \ln 2) \geq 1.21973$ to 1.26575. (This bound was already reported in the proceedings version [18] of this paper.) This is a much stronger bound than that in [17], the structures exploited are different, and the proof is simpler.

Theorem 1.2. Let $f(s) = 2 + e^{-3s}(s+1)\left(1 - \frac{s^2}{2} - \frac{s^3}{3} - \frac{s^4}{8}\right) + e^{-2s}\left(2s + \frac{5s^2}{2} + \frac{s^3}{2}\right) - e^{-s}(3 + 2s)$, and let $\beta \approx 1.26575$ be the positive root of $f(s) - 1 = 0$. Then, $C_{HAM} \geq \beta$.

1.4. **Further Related Works.** 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 *a.a.s.* one can construct H in less than $n^{(d-1)/d}\omega$ rounds where $d \geq 2$ is the degeneracy of G and $\omega = \omega(n)$ is any function that tends to infinity as $n \to \infty$. This property was recently revisited in [2], where a conjecture from [5] was proven for any graph H: *a.a.s.* it takes at least $n^{(d-1)/d}/\omega$ rounds to create H. The property of having cliques of order tending to infinity as $n \to \infty$ was investigated in [16]. In [22], k-factors and k-connectivity were studied.

Another property studied in the context of semi-random processes is that of having a perfect matching, which we denote by PM. Since the 2-out process has a perfect matching *a.a.s.* [28], and the semi-random process can simulate the 2-out process, we immediately get that $C_{PM} \leq 2$. By simulating the semi-random process with another random graph process known to have a perfect matching *a.a.s.* [21], the bound can be improved to $1 + 2/e < 1.73576$ [5]. This bound was recently improved by investigating another fully adaptive algorithm [19], giving the current best bound of $C_{PM} < 1.20524$. The same paper improves the lower bound observed in [5] of $C_{PM} \geq \ln(2) > 0.69314$ to $C_{PM} > 0.93261$. While the optimal value of C_{PM} remains unknown, a general purpose theorem is proven in [24] that identifies a sufficient condition for a property $\mathcal{P}$ to have a sharp threshold. When $\mathcal{P}$ is HAM or PM, [24] uses the theorem to establish the existence of sharp thresholds for these properties.

Let us now discuss what is known about the property of containing a given spanning graph H as a subgraph. It was asked by Noga Alon whether for any bounded-degree H, one can construct a copy of H *a.a.s.* in $O(n)$ rounds. This question was answered positively in a strong sense in [4], in which it was shown that any graph with maximum degree Δ can be constructed *a.a.s.* in $(3\Delta/2 + o(\Delta))n$ rounds and, if $\Delta = \omega(\log(n))$, in $(\Delta/2 + o(\Delta))n$ rounds. Note that these upper bounds are asymptotic in Δ; when Δ is constant in n, such as for perfect matchings and Hamiltonian cycles, they give no concrete bound.

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 [11], where in each

---

[1]`https://math.torontomu.ca/~pralat/`

round, a random spanning tree of K_n is presented to the algorithm, who chooses one of the edges to keep. In [26], k random vertices rather than just one are offered, and the algorithm chooses one of them before creating an edge. In [24], 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 [11] and [26]. In [20], the vertices offered by the process follow a random permutation. Finally, hypergraphs are investigated in [2, 25, 3].

2. Proof of Theorem 1.1

2.1. **Algorithmic Preliminaries.** In this section, we introduce some notation/terminology as well as the basic ideas used in the design of all of our strategies. We say that vertex $x \in [n]$ is *covered* by u_t arriving at round t, or that u_t *lands* on x, provided $u_t = x$. The main ingredient for proving Theorem 1.1 is to specify a strategy which keeps "extending" or "augmenting" a path P – as will be explained momentarily – as well as building a collection $\mathcal{Y}$ of edges, all vertex-disjoint from one another and P, until all edges in $\mathcal{Y}$ are joined to P and P becomes Hamiltonian. Then, with a few more steps (just $o(n)$), the Hamiltonian path P can be completed into a Hamiltonian cycle.

Suppose that after $t \geq 0$ steps, we have constructed the graph G_t which contains the path P_t and the collection $\mathcal{Y}_t$ of disjoint edges. Let $V(P_t)$ and $V(\mathcal{Y}_t)$ denote respectively the vertices in P_t and in $\mathcal{Y}_t$, and U_t the vertices in neither. Denote the (induced) distance between vertices $x, y \in V(P_t)$ on the path P_t by $d_{P_t}(x, y)$. We also define $d_{P_t}(x, Q) := \min_{q \in Q} d_{P_t}(x, q)$, for $x \in V(P_t)$ and $Q \subseteq V(P_t)$. In step $t + 1$, $u_{t+1} \in U_t, V(\mathcal{Y}_t)$, or $V(P_t)$:

- If $u_{t+1} \in U_t$, we extend the collection $\mathcal{Y}_t$ by choosing v_{t+1} to be a different vertex in U_t and adding $u_{t+1}v_{t+1}$ to $\mathcal{Y}_t$. (If $|U_t| = 1$, we simply "pass" on the round, choosing v_{t+1} arbitrarily and not using $u_{t+1}v_{t+1}$ for the construction of the Hamiltonian cycle. However, this is unlikely to happen until the very end of the process, when we apply a different strategy; see Section 2.4.) We call such a move a *(greedy) $\mathcal{Y}$-extension*.
- If $u_{t+1} \in V(\mathcal{Y}_t)$, an extension of P_t can be made by appending the $\mathcal{Y}_t$ edge on u_{t+1} to an end of P_t, and deleting it from $\mathcal{Y}_t$. Such a move is called a *(greedy) path extension*.
- If $u_{t+1} \in V(P_t)$ we cannot perform a greedy path extension, but we can still choose v_{t+1} in a way that will help us extend the path in future rounds. Specifically, choose $v_{t+1} \in U_t \cup V(\mathcal{Y}_t)$ to be an isolated vertex or an edge endpoint; it could be chosen uniformly at random, though we will use a more efficient strategy. Consider a future round $i > t$ where u_{i+1} happens to be a path neighbour of u_{t+1} (i.e., $d_{P_i}(u_{t+1}, u_{i+1}) = 1$). In this case, if $v_{t+1} \in U_i$ is an isolated vertex, set $v_{i+1} = v_{t+1}$, and replace the path edge $\{u_{t+1}, u_{i+1}\}$ with the length-2 path $(u_{t+1}, v_{t+1} = v_{i+1}, u_{i+1})$, thus making P one edge longer. If $v_{t+1} \in V(\mathcal{Y}_i)$ belongs to an isolated edge, choose v_{i+1} to be its neighbour, and replace the path edge $\{u_{t+1}, u_{i+1}\}$ with the length-3 path $(u_{t+1}, v_{t+1}, v_{i+1}, u_{i+1})$, making P two edges longer. Call either of these cases a *path augmentation*.

2.2. **Proof Overview.** In order to prove Theorem 1.1, we analyze a strategy which proceeds in three distinct *stages*. In the first stage, we execute `DegreeGreedy`, an algorithm which makes greedy $\mathcal{Y}$-extensions and path extensions whenever possible, and otherwise sets up path augmentation operations for future rounds in a degree-greedy manner. During the execution of `DegreeGreedy` some edges are coloured red or blue to help keep track of when these augmentations can be made. We use two colours, namely red and blue, to distinguish between edges which are added randomly (red) and greedily (blue). In step $t + 1$, v_{t+1} is chosen amongst $U_t \cup V(\mathcal{Y}_t)$ that are incident with the least number of blue edges. This degree-greedy decision is done to minimize the number of coloured vertices which are destroyed when path augmentations and extensions are made in later rounds. This stage lasts for N *phases*, where N is any non-negative integer that may be viewed as the parameter of the algorithm (here a phase is a contiguous set of steps shorter than the full

stage). For the claimed (numerical) upper bound of Theorem 1.1, N is set to 100. Setting smaller values of the parameter N—in particular, setting $N = 0$—yields an algorithm that is easier to analyse. Setting $N > 100$ can slightly improve the bound in Theorem 1.1, but the gain is rather insignificant.

The output of the first phase are P and $\mathcal{Y}$ that have been constructed, together with the set $\mathcal{E}$ of red edges (all blue edges will be discarded). The second stage takes $(P, \mathcal{Y}, \mathcal{E})$ as input, and executes a procedure called `FullyRandomized`, an algorithm which makes greedy $\mathcal{Y}$-extensions or path extensions whenever possible, and otherwise chooses v_{t+1} randomly amongst $U_t \cup V(\mathcal{Y}_t)$. We execute `FullyRandomized` until we are left with εn vertices in $U_t \cup V(\mathcal{Y}_t)$, where $\varepsilon = \varepsilon(n)$ tends to 0 as $n \to \infty$ arbitrarily slowly. (In practice, one can set ε to be an arbitrarily small positive number when running this algorithm.) At this point, we proceed to the final stage where a clean-up algorithm is run, which uses merely path augmentations. Using well-known concentration inequalities we prove that a Hamiltonian cycle can be constructed in an additional $O(\sqrt{\varepsilon} n) = o(n)$ steps.

In Section 2.3, we first describe `FullyRandomized`, as it is easier to state and analyze than `DegreeGreedy`. Moreover, if we take $N = 0$, which corresponds to executing `FullyRandomized` from the beginning, then we will be left with a path on all but εn vertices after $\alpha^* n$ steps where $\alpha^* \leq 1.84887$. This is exactly the upper bound obtained in [15]. Our third stage clean-up algorithm from Section 2.4 allows us to complete the Hamiltonian cycle in another $o(n)$ steps. Thus, Sections 2.3 and 2.4 provide a self-contained proof of an upper bound on $C_{\texttt{HAM}}$ of $\alpha^* \leq 1.84887$ (see Theorem 2.6). Afterwards, in Section 2.5 we formally state and analyze our first stage algorithm. This is the most technical section of the paper, as `DegreeGreedy` makes decisions in a more intelligent manner than `FullyRandomized` which necessitates more random variables in its analysis. By executing these three stages in the aforementioned order, we attain the claimed upper bound of Theorem 1.1.

2.3. **A Fully Randomized Algorithm.** The algorithm takes a tuple $(P, \mathcal{Y}, \mathcal{E})$ as input where

- P is a path on a subset of vertices in $[n]$;
- $\mathcal{Y}$ is a set of pairs of vertices in $[n] \setminus V(P)$;
- $\mathcal{E}$ is a set of *red* edges; each red edge has exactly one endpoint on P and no two edges in $\mathcal{E}$ are adjacent to the same vertex in P; moreover, these endpoints are at distance at least 3 on the path from each other.

In order to simplify the analysis, we begin the semi-random graph process from round $t = 0$ with the initial graph G_0 induced by $(P, \mathcal{Y}, \mathcal{E})$. Note that if $N = 0$, then G_0 is the empty graph on $[n]$. We encourage the reader to keep this case in mind on a first read through.

When considering G_t, a certain subset of its edges will be coloured red. This helps us define certain vertices used by our algorithm for path augmentations. A vertex $x \in V(P_t)$ is *one-red* provided it is adjacent to precisely one red edge of G_t. Similarly, $x \in V(P_t)$ is *two-red*, provided it is adjacent to precisely two red edges of G_t. Throughout the execution of `FullyRandomized`, each vertex in $V(P_t)$ is incident with at most two red edges. We denote the sets of one-red vertices and two-red vertices by $\mathcal{L}_t^1$ and $\mathcal{L}_t^2$, respectively, and refer to $\mathcal{L}_t := \mathcal{L}_t^1 \cup \mathcal{L}_t^2$ as the *red* vertices of G_t. By definition, $\mathcal{L}_t^1$ and $\mathcal{L}_t^2$ are disjoint. Initially, $P_0 = P$, $\mathcal{Y}_0 = \mathcal{Y}$, $\mathcal{L}_0^2 = \emptyset$, and $\mathcal{L}_0^1$ is set to be the set of vertices in $V(P)$ that are incident with a red edge in $\mathcal{E}$. It will also be convenient to maintain a set of *permissible vertices* $\mathcal{Q}_t \subseteq V(P_t)$ which specifies which uncoloured vertices on the path can be turned red. In order to simplify our analysis, we specify the size of $\mathcal{Q}_t$ and ensure that it only contains vertices of path distance at least 3 from the red vertices on P_t. Formally:

(i) $|\mathcal{Q}_t| = \max\{|V(P_t)| - 5|\mathcal{L}_t|, 0\}$.
(ii) If $\mathcal{L}_t \neq \emptyset$, then each $x \in \mathcal{Q}_t$ satisfies $d_{P_t}(x, \mathcal{L}_t) \geq 3$.

When $\mathcal{L}_t = \emptyset$, we simply take $\mathcal{Q}_t = V(P_t)$. Otherwise, since $|\{x \in V(P_t) : d_{P_t}(x, \mathcal{L}_t) \leq 2\}| \leq 5|\mathcal{L}_t|$, we can maintain these properties by initially taking $\{x \in V(P_t) : d_{P_t}(x, \mathcal{L}_t) \geq 3\}$, and then (if

needed) arbitrarily removing $|\{x \in V(P_t) : d_{P_t}(x, \mathcal{L}_t) \geq 3\}| - \max\{|V(P_t)| - 5|\mathcal{L}_t|, 0\}$ vertices from it.

Upon the arrival of u_{t+1}, there are five main cases our algorithm must handle. The first three cases involve extending P_t or $\mathcal{Y}_t$, whereas the latter two describe what to do when it is not possible to extend the path in the current round, and how the one-red and two-red vertices are created.

(1) If u_{t+1} lands within U_t, then choose v_{t+1} u.a.r., and greedily extend $\mathcal{Y}_t$ unless $|U_t| = 1$.
(2) If u_{t+1} lands in $V(\mathcal{Y}_t)$, then greedily extend P_t.
(3) If u_{t+1} lands at path distance one from some $x \in \mathcal{L}_t$, then augment P_t via an arbitrary red edge of x.
(4) If u_{t+1} lands in $\mathcal{Q}_t$, then choose v_{t+1} u.a.r. amongst $U_t \cup V(\mathcal{Y}_t)$, and colour $u_{t+1}v_{t+1}$ red. This case creates a one-red vertex.
(5) If u_{t+1} lands in $\mathcal{L}_t^1$, then choose v_{t+1} u.a.r. amongst U_t and colour $u_{t+1}v_{t+1}$ red. This case converts a one-red vertex to a two-red vertex.

In all the remaining cases, we choose v_{t+1} arbitrarily, and interpret the algorithm as *passing* on the round, meaning the edge $u_{t+1}v_{t+1}$ will not be used to construct a Hamiltonian cycle. In particular, the algorithm passes on rounds in which u_{t+1} lands at path distance two from some $x \in \mathcal{L}_t$. This guarantees that no two red vertices are at distance two from each other and so when u_{t+1} lands next to a red vertex, this neighbouring red vertex is uniquely identified. Let us say that a red vertex is *well-spaced*, provided it is at distance at least 3 on the path from all other red vertices, and it is *not* an endpoint of P_t. Observe that each well-spaced red vertex yields precisely two vertices on P_t where a path augmentation involving u_{t+1} can occur. By construction, all but at most 2 of the algorithm's red vertices are well-spaced. The step $t+1$ of the algorithm when u_{t+1} is drawn u.a.r. from $[n]$ is formally described by the `FullyRandomized` algorithm. Specifically, we describe how the algorithm chooses v_{t+1}, how it constructs P_{t+1}, and how it adjusts the colours of G_{t+1}, thus updating $\mathcal{L}_t^1$ and $\mathcal{L}_t^2$.

We define the random variables $X(t) = |V(P_t)|$, $L_1(t) = |\mathcal{L}_t^1|$, $L_2(t) = |\mathcal{L}_t^2|$, $L(t) = |\mathcal{L}_t| = L_1(t) + L_2(t)$, and $Y(t) = |V(\mathcal{Y}_t)| = 2|\mathcal{Y}_t|$. Note that $L(t)$ is an auxiliary random variable which we define only for convenience, and $Y(t)$ denotes the number of vertices incident to edges in $\mathcal{Y}_t$.

The input $(P, \mathcal{Y}, \mathcal{E})$ of `FullyRandomized` is the output of `DegreeGreedy`, and thus is randomized. Our analysis of the execution of `FullyRandomized` relies on the fact that $(P, \mathcal{Y}, \mathcal{E})$ has a certain distribution. To be specific, recall that $\mathcal{L}_0^1$ is the set of vertices on P incident with an edge in $\mathcal{E}$. Then, conditional on P, $|\mathcal{Y}|$, and $\mathcal{L}_0^1$, the following properties are satisfied by $(P, \mathcal{Y}, \mathcal{E})$:

(O_1) $\mathcal{Y}$ is uniform over all possible $|\mathcal{Y}|$ pairs of vertices in $[n] \setminus V(P)$;
(O_2) $\mathcal{E}$ is uniform over all possible set of edges joining $\mathcal{L}_0^1$ and $[n] \setminus V(P)$ such that every vertex in $\mathcal{L}_0^1$ is incident with exactly one edge in $\mathcal{E}$.

We shall prove that these properties hold in Subsection 2.7. Using these properties, together with the specification of `FullyRandomized`, we first show that our random variables cannot change drastically in one round. We use Δ to denote the one step changes in our random variables (i.e., $\Delta X(t) := X(t+1) - X(t)$).

Lemma 2.1 (Boundedness Hypothesis – `FullyRandomized`). *With probability $1 - O(n^{-1})$,*

$$\max\{|\Delta X(t)|, |\Delta L_1(t)|, |\Delta L_2(t)|, |\Delta Y(t)|\} = O(\log n)$$

for all $0 \leq t \leq 3n - |\mathcal{E}|$ with $n - X(t) \geq n/\log n$.

Proof. Note that, by design, the path can only increase its length but it cannot absorb more than two vertices in each round. Hence, the desired property clearly holds for the random variable $X(t)$. The same holds for $Y(t)$. To estimate the maximum change for the random variables $L_1(t)$ and $L_2(t)$, we need to upper bound the number of red edges adjacent to any particular vertex $v \in U_t \cup V(\mathcal{Y}_t)$. Observe that due to (O_2) and how the red edges are randomly selected, if we condition on $U_t \cup V(\mathcal{Y}_t)$, then this is stochastically upper bounded by $\text{Bin}(t + |\mathcal{E}|, |U_t \cup V(\mathcal{Y}_t)|^{-1})$.

ALGORITHM `FullyRandomized`. Step $t+1$

1: **if** $u_{t+1} \in U_t$ and $|U_t| \geq 2$ **then**   ▷ greedily extend $\mathcal{Y}_t$
2:     Let v_{t+1} be a uniformly random vertex in $U_t \setminus \{u_{t+1}\}$.
3:     Set $P_{t+1} = P_t$ and $\mathcal{Y}_{t+1} = \mathcal{Y}_t \cup \{u_{t+1}v_{t+1}\}$.
4: **else if** $u_{t+1}y \in \mathcal{Y}_t$ for some y **then**   ▷ greedily extend P_t
5:     Let v_{t+1} be an arbitrarily chosen endpoint of P_t.
6:     Set $V(P_{t+1}) = V(P_t) \cup \{u_{t+1}, y\}$, $E(P_{t+1}) = E(P_t) \cup \{u_{t+1}v_{t+1}, u_{t+1}y\}$.
7:     Set $\mathcal{Y}_{t+1} = \mathcal{Y}_t \setminus \{u_{t+1}y\}$.
8:     Uncolour all of the edges adjacent to u_{t+1} or y.
9: **else if** $d_{P_t}(u_{t+1}, \mathcal{L}_t) = 1$ **then**   ▷ path augment via red vertices
10:     Let $x \in \mathcal{L}_t$ be the (unique) red vertex adjacent to u_{t+1}
11:     Denote $xy \in E(G_t)$ an arbitrary red edge of x.
12:     **if** $y \in U_t$ **then**
13:         Set $v_{t+1} = y$.
14:         Set $V(P_{t+1}) = V(P_t) \cup \{v_{t+1}\}$ and $E(P_{t+1}) = (E(P_t) \cup \{xv_{t+1}, u_{t+1}v_{t+1}\}) \setminus \{u_{t+1}x\}$.
15:         Set $\mathcal{Y}_{t+1} = \mathcal{Y}_t$
16:         Uncolour all of the edges adjacent to r.
17:     **else if** $yy' \in \mathcal{Y}_t$ for some y' **then**
18:         Set $v_{t+1} = y'$
19:         Set $V(P_{t+1}) = V(P_t) \cup \{y, v_{t+1}\}$, $E(P_{t+1}) = (E(P_t) \cup \{xy, yv_{t+1}, u_{t+1}v_{t+1}\}) \setminus \{u_{t+1}x\}$.
20:         Set $\mathcal{Y}_{t+1} = \mathcal{Y}_t \setminus \{yv_{t+1}\}$
21:         Uncolour all of the edges adjacent to y or v_{t+1}.
22:     **end if**
23: **else**
24:     **if** $u_{t+1} \in \mathcal{Q}_t \cup \mathcal{L}_t^1$ **then**   ▷ construct red vertices or pass
25:         Choose v_{t+1} u.a.r. from $U_t \cup V(\mathcal{Y}_t)$.
26:         Colour $u_{t+1}v_{t+1}$ red.   ▷ construct a one-red or two-red vertex
27:     **else**   ▷ pass on $u_{t+1}v_{t+1}$
28:         Choose v_{t+1} arbitrarily from $[n]$.
29:     **end if**
30:     Set $P_{t+1} = P_t$; $\mathcal{Y}_{t+1} = \mathcal{Y}_t$.
31: **end if**
32: Update U_{t+1} and $\mathcal{L}_{t+1}$.
33: Update $\mathcal{Q}_{t+1}$, if needed, such that $|\mathcal{Q}_{t+1}| = |V(P_{t+1})| - 5|\mathcal{L}_{t+1}|$.

Since $t + |\mathcal{E}| \leq 3n$, and we have assumed that there are at least $n/\log n$ vertices in $U_t \cup V(\mathcal{Y}_t)$, the number of red edges adjacent to v is stochastically upper bounded by the binomial random variable $\text{Bin}(3n, \log n/n)$ with expectation $3 \log n$. It follows immediately from Chernoff's bound that with probability $1 - O(n^{-3})$, the number of red edges adjacent to v is $O(\log n)$, and so the desired bound holds by union bounding over all $3n^2$ vertices and steps. □

Let us denote $H_t = (X(i), L_1(i), L_2(i), Y(i))_{0 \leq i \leq t}$. Note that H_t does *not* encompass the entire history of the random process after t rounds (i.e., $G_0, \ldots, G_t$, the first $t + 1$ graphs appearing in the sequence generated by the process). The distribution of $(P, \mathcal{Y}, \mathcal{E})$ together with the technique of deferred information exposure permit a tractable analysis of the random positioning of v_t when u_t is red. In particular, as we only expose $Y(t)$ instead of $\mathcal{Y}_t$, $\mathcal{Y}_t$ has the same distribution (conditional on H_t) as first exposing the set of vertices in $[n] \setminus V(P_t)$, then uniformly selecting a subset of vertices in $[n] \setminus V(P_t)$ of cardinality $Y(t)$, and then finally taking a uniformly random perfect matching over the $Y(t)$ vertices (i.e. pair the $Y(t)$ vertices into $Y(t)/2$ disjoint edges). Similarly, conditional on

$L_1(t)$ and $L_2(t)$, we may first expose $\mathcal{L}_t^1$ and $\mathcal{L}_t^2$, and then choose their neighbours joined by a red edge uniformly from $[n] \setminus V(P_t)$. Moreover, this process (of choosing the ends of red edges lying in $[n] \setminus V(P_t)$) is independent of the process of choosing and pairing vertices for $\mathcal{Y}_t$. We observe the following expected difference equations.

Lemma 2.2 (Trend Hypothesis – `FullyRandomized`). *For each $t \geq 0$, if $n - X(t) \geq n / \log n$, then by setting $\Gamma(t) = 1 + Y(t)/(n - X(t))$ and $A(t) = 2Y(t)/(n - X(t))$,*

$$
\mathbb{E}[\Delta X(t) \mid H_t] = \frac{2Y(t)}{n} + \frac{2L(t)}{n} \cdot \Gamma(t) + O(\log n / n) \tag{2}
$$

$$
\mathbb{E}[\Delta Y(t) \mid H_t] = -\frac{2Y(t)}{n} + \frac{2(n - X(t) - Y(t))}{n} - \frac{2L(t)}{n} \cdot A(t) + O(\log n / n) \tag{3}
$$

$$
\mathbb{E}[\Delta L_1(t) \mid H_t] = \frac{X(t) - 5L(t)}{n} - \frac{2L_1(t)}{n} + \frac{2L_1(t)}{n} \left(\frac{2L_2(t)}{n - X(t)} - \frac{L_1(t)}{n - X(t)} \right) \cdot \Gamma(t)
$$
$$
+ \frac{2L_2(t)}{n} + \frac{2L_2(t)}{n} \left(\frac{2L_2(t)}{n - X(t)} - \frac{L_1(t)}{n - X(t)} \right) \cdot \Gamma(t) - \frac{L_1(t)}{n}
$$
$$
+ \frac{2Y(t)}{n} \cdot \left(\frac{2L_2(t)}{n - X(t)} - \frac{L_1(t)}{n - X(t)} \right) + O(\log n / n) \tag{4}
$$

$$
\mathbb{E}[\Delta L_2(t) \mid H_t] = \frac{L_1(t)}{n} - \frac{2L_2(t)}{n} \cdot A(t) - \frac{2L_1(t)}{n} \cdot \frac{2L_2(t)}{n - X(t)} \cdot B(t)
$$
$$
- \frac{2L_2(t)}{n} - \frac{2L_2(t)}{n} \cdot \frac{2L_2(t)}{n - X(t)} \cdot \Gamma(t) + O(\log n / n). \tag{5}
$$

Proof. As discussed earlier, the `FullyRandomized` algorithm ensures that at time t there are at most 2 red vertices which are not well-spaced. Thus, since our expected differences each allow for a $O(\log n / n)$ term, without loss of generality, we can assume that all our red vertices are well-spaced. Note also that all our explanations below assume that we have conditioned on H_t.

When path augmentation occurs via a red edge incident to a vertex x on P_t, we first expose r (the other end of the red edge) which is distributed uniformly over all vertices in $[n] \setminus V(P_t)$, and then we expose whether r is in $V(\mathcal{Y}_t)$. In the case that r is in $V(\mathcal{Y}_t)$ we expose y which is paired to r in $\mathcal{Y}_t$.

The first expected difference is easy to see. Observe that there are three disjoint cases where $\Delta X(t)$ is nonzero. Case 1: u_{t+1} lands on a vertex in $V(\mathcal{Y}_t)$. In this case $\Delta X(t)$ is 2, and this event occurs with probability $Y(t)/n$. Case 2: u_{t+1} is next to a red vertex x (i.e. a vertex in $\mathcal{L}_t$) on P_t, and path augmentation is performed via a red edge xr where $r \in U_t$. In this case, $\Delta X(t) = 1$ and the probability of this event is $(2L(t)/n) \cdot (1 - Y(t)/(n - X(t)))$, where $2L(t)/n$ is the probability that $d_{P_t}(u_{t+1}, \mathcal{L}_t) = 1$, and $1 - Y(t)/(n - X(t))$ is the probability that $r \in U_t$ conditional on $r \in [n] \setminus V(P_t)$. Case 3: same as case 2 but $r \in V(\mathcal{Y}_t)$. In this case, $\Delta X(t) = 2$ and the probability of this event is $(2L(t)/n) \cdot (Y(t)/(n - X(t)))$. Combining all three cases together we obtain (2).

The remaining equations are obtained in a similar manner. In what follows, we explain the event A for which each term in the equations accounts for as $\mathbb{E}[\Delta Z(t) \cdot \mathbf{1}_A]$ where $Z \in \{Y, L_1, L_2\}$.

In the second equation, $-2Y(t)/n$ is the contribution from the case where u_{t+1} lands on $V(\mathcal{Y}_t)$; $(-2L(t)/n) \cdot A(t)$ corresponds to the event that u_{t+1} lands on a neighbour of some $x \in \mathcal{L}_t$ on P_t, and the path augmentation is performed via a red edge xr where $r \in V(\mathcal{Y}_t)$. Finally, $2(n - X(t) - Y(t))/n$ corresponds to the event that u_{t+1} lands on a vertex in U_t.

In the third equation, the term $(X(t) - 5L(t))/n$ is the contribution from the case where u_{t+1} lands on $\mathcal{Q}_t$. In the case where u_{t+1} lands on a vertex neighbouring some $x \in \mathcal{L}_t^1$ on P_t (which occurs with probability $2L_1(t)/n$) the contribution to $\Delta L_1(t)$ can come from two sources: (a) x is removed from $\mathcal{L}_t^1$ after the path augmentation and thus it contributes -1 to $\Delta L_1(t)$; (b) one or two vertices will be added to P_t, which results in uncolouring of all red edges incident to them, and

which consequently contributes to $\Delta L_1(t)$. Note that $2L_2(t)/(n - X(t))$ is the expected number of two-red vertices that become one-red when a vertex $r \in U_t \cup V(\mathcal{Y}_t)$ is moved to P_t. Similarly, $L_1(t)/(n - X(t))$ is the expected number of one-red vertices that get removed from $\mathcal{L}_t^1$ due to moving a certain vertex $r \in U_t \cup V(\mathcal{Y}_t)$ to P_t. Finally, $B(t)$ is the expected number of vertices in $U_t \cup V(\mathcal{Y}_t)$ that will be added to P_t.

The next two terms (in the second line of third equation) correspond to the symmetric case: u_{t+1} lands on a vertex neighbouring some $x \in \mathcal{L}_t^2$ on P_t (which occurs with probability $2L_2(t)/n$) with the same two sources that contribute to $\Delta L_1(t)$. The term $-L_1(t)/n$ corresponds to the case where u_{t+1} lands on a vertex $x \in \mathcal{L}_t^1$ which results in moving x from $\mathcal{L}_t^1$ to $\mathcal{L}_{t+1}^2$.

Finally, the last term $(2Y(t)/n)(2L_2(t) - L_t(t))/(n - X(t))$ is the contribution from moving two vertices from $U_t \cup V(\mathcal{Y}_t)$ to P_t in the case where u_{t+1} lands on a vertex in $V(\mathcal{Y}_t)$.

For the last equation, $L_1(t)/n$ accounts for u_{t+1} landing on a vertex in $\mathcal{L}_t^1$, and a one-red vertex becomes two-red. The term $-2L_2(t)A(t)/n$ accounts for the case where u_{t+1} lands on a vertex in $V(\mathcal{Y}_t)$. In this case (which occurs with probability $Y(t)/2$) 2 vertices are moved from $V(\mathcal{Y}_t)$ to P_t, each of which will be resulting in uncolouring $2L_2(t)/(n - X(t))$ red edges incident to vertices in $\mathcal{L}_t^2$ in expectation.

The third and the fifth terms together in the equation account for the case where u_{t+1} lands on a neighbour of $\mathcal{L}_t$, where one or two vertices in $U_t \cup V(\mathcal{Y}_t)$ are moved to P_t after the path augmentation, each resulting in uncolouring $2L_2(t)/(n - X(t))$ red edges incident to $\mathcal{L}_t^2$ in expectation. The fourth term accounts for the case where u_{t+1} lands on a vertex neighbouring a vertex $x \in \mathcal{L}_t^2$, resulting in the removal of x from $\mathcal{L}_t^2$ after the path augmentation. □

In order to analyze `FullyRandomized`, we shall employ the differential equation method [30]. This method is commonly used in probabilistic combinatorics to analyze random processes that evolve step by step. The step changes must be small in relation to the entirety of the discrete structure. For instance, in our application, this refers to adding one edge at a time to the graph on n vertices. The method allows us to derive tight bounds on the associated random variables which hold a.a.s. at every step of the random process. We refer the reader to [6] for a gentle introduction to the methodology.

Recall that `FullyRandomized` takes input $(P, \mathcal{Y}, \mathcal{E})$. Let $X(0), Y(0), L_1(0)$ denote the number of vertices on P, the number of vertices incident to edges in $\mathcal{Y}$, and the number of vertices incident with $\mathcal{E}$, respectively. We prove in Section 2.7 that there exist some constants $\hat{x}, \hat{y}, \hat{\ell}_1$ such that $|X(0)/n - \hat{x}|, |Y(0)/n - \hat{y}|, |L_1(0)/n - \hat{\ell}_1| \leq \lambda$ for some $\lambda = o(1)$. Initially, there are no two-red vertices, that is, we will always set $L_2(0) = 0$. Let us now fix a sufficiently small constant $\varepsilon > 0$, and define the bounded domain

$$\mathcal{D}_\varepsilon := \{(s, x, y, \ell_1, \ell_2) : -1 < s < 3, -1 < x < 1 - \varepsilon, |\ell_1| < 2, |\ell_2| < 2\}.$$

Consider the system of differential equations in variable s with functions $x = x(s), y = y(s), \ell_1 = \ell_1(s)$, and $\ell_2 = \ell_2(s)$:

$$x' = 2y + 2(\ell_1 + \ell_2)\lambda \tag{6}$$

$$y' = -2y + 2(1 - x - y) - 2(\ell_1 + \ell_2)a \tag{7}$$

$$\ell_1' = x - 5(\ell_1 + \ell_2) - 2\ell_1 + (2\ell_1\lambda + 2\ell_2\lambda + 2y) \cdot \frac{2\ell_2 - \ell_1}{1 - x} + 2\ell_2 - \ell_1 \tag{8}$$

$$\ell_2' = \ell_1 - 2\ell_2 a - (2\ell_1 + 2\ell_2)\lambda \cdot \frac{2\ell_2}{1 - x} - 2\ell_2, \tag{9}$$

where $\lambda(s) = 1 + y(s)/(1 - x(s))$ and $a(s) = 2y(s)/(1 - x(s))$. The right-hand side of each of the above equations is Lipchitz on the domain $\mathcal{D}_\varepsilon$. Define

$$T_{\mathcal{D}_\varepsilon} = \min\{t \geq 0 : (t/n, X(t)/n, Y(t)/n, L_1(t)/n, L_2(t)/n) \notin \mathcal{D}_\varepsilon\}.$$

Now, the 'Initial Condition' of Theorem A.1 is satisfied with values $(0, \hat{x}, \hat{y}, \hat{\ell}_1, 0)$ and some $\lambda = o(1)$. Moreover, the 'Trend Hypothesis' and 'Boundedness Hypothesis' are satisfied with some $\delta = O(\log n/n)$, $\beta = O(\log n)$ (with failure probability $\gamma = o(1)$ throughout the process) by Lemmas 2.1 and 2.2. Thus, for every $\delta > 0$, $X(t) = nx(t/n) + o(n)$, $Y(t) = ny(t/n) + o(n)$, $L_1(t) = n\ell_1(t/n) + o(n)$ and $L_2(t) = n\ell_2(t/n) + o(n)$ uniformly for all $t_0 \leq t \leq (\sigma(\varepsilon) - \delta)n$, where x, y, ℓ_1 and ℓ_2 are the unique solution to (6)–(9) with initial conditions $x(0) = \hat{x}$, $y(0) = \hat{y}$, $\ell_1(0) = \hat{\ell}_1$, and $\ell_2(0) = 0$, and $\sigma(\varepsilon)$ is the supremum of s to which the solution can be extended before reaching the boundary of $\mathcal{D}_\varepsilon$.

Lemma 2.3 (Concentration of `FullyRandomized`'s Random Variables). *For every $\delta > 0$, a.a.s. for all $0 \leq t \leq (\sigma(\varepsilon) - \delta)n$,*

$$\max\left\{ |X(t) - x(t/n)n|, |Y(t) - y(t/n)n|, |L_1(t) - \ell_1(t/n)n|, |L_2(t) - \ell_2(t/n)n| \right\} = o(n).$$

As $\mathcal{D}_\varepsilon \subseteq \mathcal{D}_{\varepsilon'}$ for every $\varepsilon > \varepsilon'$, $\sigma(\varepsilon)$ is monotonically nondecreasing as $\varepsilon \to 0$. Thus,

$$\alpha^* := \lim_{\varepsilon \to 0+} \sigma(\varepsilon) \tag{10}$$

exists. It is clear that $|L_1(t)/n|$, $|L_2(t)/n|$, and $|Y(t)/n|$ are all bounded by 1 for all t and thus, when t/n approaches α^*, either $X(t)/n$ approaches 1 or t/n approaches 3. Formally, we have the following proposition.

Proposition 2.4. *For every $\varepsilon > 0$, there exists $\delta > 0$ such that a.a.s. one of the following holds.*
- $X(t) > (1 - \varepsilon)n$ *for all* $t \geq (\alpha^* - \delta)n$;
- $\alpha^* = 3$.

The ordinary differential equations (6)–(9) do not have an analytical solution. In both cases, $N = 0$ and $N = 100$, numerical solutions show that $\alpha^* < 1.85$. (For $N = 0$, $\alpha^* \approx 1.84887$.) Thus, by the end of the execution of `FullyRandomized`, there are εn unsaturated vertices (i.e. vertices not in P_t) remaining, for some $\varepsilon = o(1)$.

2.4. **A Clean-up Algorithm.** Suppose that we are presented a path P on $(1-\varepsilon)n$ vertices of $[n]$, where $0 < \varepsilon = \varepsilon(n) < 1/1000$. The assumption on ε is a mild but convenient one. We will apply the argument for $\varepsilon = o(1)$. In this section, we provide an algorithm for the semi-random graph process which absorbs the remaining εn vertices into P to form a Hamiltonian path, after which a Hamiltonian cycle can be constructed. The whole procedure takes $O(\sqrt{\varepsilon} n + n^{3/4} \log^2 n) = o(n)$ further steps in the semi-random graph process. Moreover, the algorithm is self-contained in that it only uses the edges of P in its execution.

Lemma 2.5 (Clean-up Algorithm). *Let $0 < \varepsilon = \varepsilon(n) < 1/1000$, and suppose that P is a path on $(1 - \varepsilon)n$ vertices of $[n]$. Then, given P initially, there exists a strategy for the semi-random graph process which builds a Hamiltonian cycle from P in $O(\sqrt{\varepsilon} n + n^{3/4} \log^2 n)$ steps a.a.s.*

Remark 2. The constant hidden in the $O(\cdot)$ notation does not depend on ε. The strategy used in the clean-up algorithm is similar to that in `FullyRandomized` but the analysis is done in a much less accurate way, as we only need to prove an $o(n)$ bound on the number of steps required to absorb εn vertices, assuming $\varepsilon = \varepsilon(n) \to 0$ as $n \to \infty$.

Proof of Lemma 2.5. Let $j_0 = \varepsilon n$. For each $k \geq 1$, let $j_k = (1/2)j_{k-1}$ if $j_{k-1} > n^{1/4}$, and let $j_k = j_{k-1} - 1$ otherwise. Clearly, j_k is a decreasing function of k. Let τ_1 be the smallest natural number k such that $j_k \leq n^{1/4}$. Let τ be the natural number k such that $j_k = 0$. It is easy to check that $\tau_1 = O(\log n)$ and $\tau = O(n^{1/4})$.

We use a clean-up algorithm, which runs in iterations. The k-th iteration repeatedly absorbs $j_{k-1} - j_k$ vertices into P, leaving j_k unsaturated vertices (vertices that have not been added to P) in the end. The k-th iteration of the clean-up algorithm works as follows.

(i) (*Initialising*): Uncolour all vertices in the graph;

(ii) (*Building reservoir*): Let $m_k := \sqrt{\varepsilon}(1/2)^{k/2}n$ for $k \le \tau_1$ and $m_k := \sqrt{n}$ if $\tau_1 < k \le \tau$. Add m_k semi-random edges as follows. If u_t lands on an unsaturated vertex, a red vertex, or a neighbour of a red vertex in P, then let v_t be chosen arbitrarily. The edge $u_t v_t$ will not be used in our construction. Otherwise, colour u_t red and choose an arbitrary v_t among those unsaturated vertices with the minimum number of red neighbours. Colour $u_t v_t$ red. Note that each red vertex is adjacent to exactly one red edge;

(iii) (*Absorbing via path augmentations*): Add semi-random edges as follows. Suppose that u_t lands on P and at least one neighbour of u_t on P is red. (Otherwise, v_t is chosen arbitrarily, and this edge will not be used in our construction.) Let x be such a red vertex (if u_t has two neighbours on P that are red, then select one of them arbitrarily). Let y by the neighbour of x such that xy is red, and let $v_t = y$. Extend P by deleting the edge xu_t and adding the edges xy and yu_t. Uncolour all red edges incident to y and all red neighbours of y (which, of course, includes vertex x).

Notice that, in each iteration, $m_k \ge n^{1/2}$. Indeed, this is true for $\tau_1 < k \le \tau$. On the other hand, if $k \le \tau_1$, then $j_k = \varepsilon n(1/2)^k$ and so $m_k = \sqrt{nj_k} \ge \sqrt{n}$ (in fact, $m_k = \Omega(n^{5/8})$).

Let T_k denote the length of the k-th iteration of the clean-up algorithm. It remains to prove that a.a.s. $\sum_{k \le \tau} T_k = O(\sqrt{\varepsilon}n + n^{3/4}\log^2 n)$. Let R_k be the number of red vertices obtained after step (ii) of iteration k. Clearly, $R_k \le m_k$. On the other hand, each u_t is coloured red with probability at least $1 - j_{k-1}/n - 3m_k/n \ge 1 - \varepsilon - 3\sqrt{\varepsilon} \ge 0.95$. Hence, R_k can be stochastically lower bounded by the binomial random variable $\text{Bin}(m_k, 0.95)$. By the Chernoff bound, with probability at least $1 - n^{-1}$, $R_k \ge 0.9m_k$, as $m_k \ge n^{1/2}$.

First, we consider iterations $k \le \tau_1$. Let $\tilde{R}_k$ be the number of red vertices at the end of step (iii). Note that the minimum degree property of step (ii) ensures that each unsaturated vertex is adjacent to at most $R_k/j_{k-1}+1 \le m_k/j_{k-1}+1$ red vertices. Moreover, exactly $j_{k-1} - j_k = (1/2)j_{k-1}$ vertices are absorbed in step (iii). As a result,

$$\tilde{R}_k \ge R_k - \left(\frac{m_k}{j_{k-1}} + 1\right) \cdot \frac{j_{k-1}}{2} \ge 0.9m_k - \frac{m_k}{2} - \frac{j_{k-1}}{2} \ge 0.3m_k,$$

as $j_{k-1} = 2j_k \le 2\sqrt{\varepsilon}m_k \le 0.1m_k$. It follows that throughout step (iii), there are at least $0.3m_k$ red vertices. Thus, for each semi-random edge added to the graph during step (iii), the probability that a path extension can be performed is at least $0.3m_k/n = 0.3\sqrt{\varepsilon}(1/2)^{k/2}$. Again, by the Chernoff bound, with probability at least $1 - n^{-1}$, the number of semi-random edges added in step (iii) is at most

$$2(j_{k-1} - j_k) \cdot \frac{2^{k/2}}{0.3\sqrt{\varepsilon}} \le 7\sqrt{\varepsilon}(1/2)^{k/2}n.$$

Combining the number of semi-random edges added in step (ii), it follows that with probability at least $1 - n^{-1}$, $T_k \le m_k + 7\sqrt{\varepsilon}(1/2)^{k/2}n = 8\sqrt{\varepsilon}(1/2)^{k/2}n$.

Next, consider iterations $\tau_1 < k \le \tau$. In each iteration, exactly one unsaturated vertex gets absorbed. The number of semi-random edges added in step (ii) is $m_k = n^{1/2}$. We have argued above that with probability at least $1 - n^{-1}$, $R_k \ge 0.9m_k$. Thus, for each semi-random edge added to the graph, the probability that a path extension can be performed is at least $0.9m_k/n = 0.9n^{-1/2}$. By the Chernoff bound, with probability at least $1 - n^{-1}$, the number of semi-random edges added in step (iii) is at most $n^{1/2}\log^2 n$. Thus, with probability at least $1 - n^{-1}$, $T_k \le n^{1/2} + n^{1/2}\log^2 n \le 2n^{1/2}\log^2 n$.

By taking the union bound over all $k \le \tau$, since $\tau = O(n^{1/4})$, it follows that a.a.s.

$$\sum_{k \le \tau} T_k \le \sum_{k \le \tau_1} 8\sqrt{\varepsilon}(1/2)^{k/2}n + \sum_{\tau_1 < k \le \tau} 2n^{1/2}\log^2 n = O(\sqrt{\varepsilon}n + n^{3/4}\log^2 n).$$

We have shown that a.a.s. by adding $O(\sqrt{\varepsilon}n + n^{3/4}\log^2 n)$ additional semi-random edges we can construct a Hamiltonian path P. To complete the job and turn it into a Hamiltonian cycle, let u and v denote the left and, respectively, the right endpoint of P. We proceed in two stages. In the first stage, add $n^{1/2}\log n$ semi-random edges $u_t v_t$ where v_t is always u, discarding any multiple edges that could possibly be created. For each such semi-random edge $u_t u$, colour the left neighbour of u_t on P blue. In the second stage, add $n^{1/2}\log n$ semi-random edges $u_t v_t$ where v_t is always v. Suppose that some $u_t = x$ is blue in the second stage. Then, a Hamiltonian cycle is obtained by deleting xy from P and adding the edges xv and uy, where y is the right neighbour of x on P. By the Chernoff bound, a.a.s. a semi-random edge added during the second stage hits a blue vertex, completing the proof. □

By setting $N = 0$ we immediately get an algorithm which a.a.s. constructs a Hamiltonian cycle in $\hat{\alpha}n$ steps, where $\hat{\alpha} \leq 1.84887$. To obtain the better bound in Theorem 1.1, we set $N = 100$, and the execution of `DegreeGreedy` will be analysed in the next subsection.

Theorem 2.6. $C_{\texttt{HAM}} \leq \hat{\alpha} \leq 1.84887$, where $\hat{\alpha}$ is defined in (10) with initial conditions for (6)–(9) set by $x(0) = y(0) = \ell_1(0) = \ell_2(0) = 0$.

Proof. This follows from Proposition 2.4, the numerical value of α^*, and Lemma 2.5.   □

2.5. **A Degree-Greedy Algorithm.** Let us suppose that after $t \geq 0$ steps, we have constructed the graph G_t which contains the path P_t and a collection of vertex disjoint edges $\mathcal{Y}_t$ where $V(\mathcal{Y}_t) \subseteq [n]\setminus V(P_t)$. We refer to $V(P_t)$ (respectively, $[n]\setminus V(P_t)$) as the *saturated* (respectively, *unsaturated*) vertices of $[n]$.

As before, our algorithm uses path augmentations, and we colour the edges and vertices of G_t to help keep track of when these augmentations can be made. We now use two colours, namely red and blue, to distinguish between edges which are added randomly (red) and greedily (blue). Our blue edges will be chosen so as to minimize the number of blue edges destroyed by path augmentations in future rounds.

We say that $x \in V(P_t)$ is *blue*, provided it is adjacent to a single blue edge of G_t, and no red edge. Similarly, $x \in V(P_t)$ is *red*, provided it is adjacent to a single red edge of G_t, and no blue edge. Finally, we say that $x \in V(P_t)$ is *magenta (mixed)*, provided it is adjacent to a single red edge, and a single blue edge. We denote the blue vertices, red vertices, and magenta (mixed) vertices by $\mathcal{B}_t, \mathcal{R}_t$ and $\mathcal{M}_t$, respectively, and define $\mathcal{L}_t := \mathcal{B}_t \cup \mathcal{R}_t \cup \mathcal{M}_t$ to be the *coloured* vertices. By definition, $\mathcal{B}_t, \mathcal{R}_t$ and $\mathcal{M}_t$ are disjoint. It will be convenient to once again define U_t as the vertices not in P_t or any edge of $\mathcal{Y}_t$. Finally, we maintain a set of *permissible* vertices $\mathcal{Q}_t$ which indicate which vertices of the path are allowed to be coloured blue. Specifically, using the same reasoning as before, we ensure the following:

   (i) $|\mathcal{Q}_t| = \max\{|V(P_t)| - 5|\mathcal{L}_t|, 0\}$.
   (ii) If $\mathcal{L}_t \neq \emptyset$, then each $x \in \mathcal{Q}_t$ satisfies $d_{P_t}(x, \mathcal{L}_t) \geq 3$.

Upon the arrival of u_{t+1}, there are six main cases our algorithm must handle. The first three cases involve extending P_t or $\mathcal{Y}_t$, whereas the latter three describe how to add edges so that the path can be extended in later rounds.

   (1) If u_{t+1} lands in U_t and $|U_t| \geq 2$, then choose v_{t+1} u.a.r. amongst $U_t \setminus \{u_{t+1}\}$ and extend $\mathcal{Y}_t$.
   (2) If u_{t+1} lands in $V(\mathcal{Y}_t)$, then greedily extend P_t.
   (3) If u_{t+1} lands at path distance one from $x \in \mathcal{L}_t$, then augment P_t via a coloured edge of x, where a blue edge is taken over a red edge if possible.
   (4) If u_{t+1} lands in $\mathcal{Q}_t$, then choose v_{t+1} u.a.r. amongst those vertices of U_t with *minimum* blue degree. The edge $u_{t+1}v_{t+1}$ is then coloured blue, and a single blue vertex is created.
   (5) If u_{t+1} lands in $\mathcal{R}_t$, then choose v_{t+1} u.a.r. amongst those vertices of U_t with minimum blue degree. The edge $u_{t+1}v_{t+1}$ is then coloured blue, and a single red vertex is converted to a magenta (mixed) vertex.

(6) If u_{t+1} lands in $\mathcal{B}_t$, then choose v_{t+1} u.a.r. amongst U_t and colour $u_{t+1}v_{t+1}$ red. This case converts a blue vertex to a magenta vertex.

In all the remaining cases, we choose v_{t+1} arbitrarily, and interpret the algorithm as *passing* on the round. As in `FullyRandomized`, we ensure that all of the algorithm's coloured vertices are at path distance at least 3 from each other, and we define a coloured vertex to be *well-spaced* in the same way. Step $t+1$ of the algorithm when u_{t+1} is drawn u.a.r. from $[n]$ is formally described by the `DegreeGreedy` algorithm. We describe how the algorithm chooses v_{t+1}, how it constructs P_{t+1} and $\mathcal{Y}_{t+1}$, and how it adjusts the colours of G_{t+1}, thus updating $\mathcal{B}_t, \mathcal{M}_t$ and $\mathcal{R}_t$. Note that red vertices

---

ALGORITHM `DegreeGreedy`. Step $t+1$

1: **if** $u_{t+1} \in U_t$ and $|U_t| \geq 2$ **then**   ▷ greedily extend $\mathcal{Y}_t$
2:      Choose v_{t+1} u.a.r. from $U_t \setminus V(\mathcal{Y}_t) \cup \{u_{t+1}\}$.
3:      Set $P_{t+1} = P_t$ and $\mathcal{Y}_{t+1} = \mathcal{Y}_t \cup \{u_{t+1}v_{t+1}\}$.
4: **else if** $u_{t+1}y \in \mathcal{Y}_t$ for some y **then**   ▷ greedily extend the path
5:      Let v_{t+1} be an arbitrarily chosen endpoint of P_t.
6:      Update P_{t+1} from P_t by adding edges $u_{t+1}v_{t+1}$ and $u_{t+1}y$.
7:      Set $\mathcal{Y}_{t+1} = \mathcal{Y}_t \setminus \{u_{t+1}y\}$.
8:      Uncolour all of the edges adjacent to u_{t+1} or y.
9: **else if** $d(u_{t+1}, \mathcal{L}_t) = 1$ **then**   ▷ path augment via coloured vertices
10:     Let $x \in \mathcal{L}_t$ be the (unique) coloured vertex adjacent to u_{t+1}
11:     **if** x is red **then**
12:         Denote $xy \in E(G_t)$ the red edge of x.
13:     **else**   ▷ x is blue or magenta
14:         Denote $xy \in E(G_t)$ the blue edge of x.
15:     **end if**
16:     **if** $y \in U_t$ **then**
17:         Set $v_{t+1} = y$
18:         Update P_{t+1} from P_t by adding edges $u_{t+1}v_{t+1}, v_{t+1}x$ and removing edge $u_{t+1}x$.
19:         Set $\mathcal{Y}_{t+1} = \mathcal{Y}_t$.
20:     **else if** $yy' \in \mathcal{Y}_t$ **then**
21:         Set $v_{t+1} = y'$
22:         Update P_{t+1} from P_t by adding edges $u_{t+1}v_{t+1}, v_{t+1}y, yx$ and removing edge $u_{t+1}x$.
23:         Set $\mathcal{Y}_{t+1} = \mathcal{Y}_t \setminus \{yy'\}$
24:     **end if**
25:     Uncolour all of the edges adjacent to y (as well as y' if applicable).
26: **else**   ▷ construct coloured vertices or pass
27:     **if** $u_{t+1} \in \mathcal{Q}_t \cup \mathcal{R}_t$ **then**
28:         Choose v_{t+1} u.a.r. from the vertices of U_t of minimum blue degree.
29:         Colour $u_{t+1}v_{t+1}$ blue.   ▷ create a blue or magenta vertex
30:     **else if** $u_{t+1} \in \mathcal{B}_t$ **then**
31:         Choose v_{t+1} u.a.r. from U_t.
32:         Colour the edge $u_{t+1}v_{t+1}$ red.   ▷ create a magenta vertex
33:     **else**   ▷ pass on using edge $u_{t+1}v_{t+1}$
34:         Choose v_{t+1} arbitrarily from $[n]$.
35:     **end if**
36:     Set $P_{t+1} = P_t, \mathcal{Y}_{t+1} = \mathcal{Y}_t$.
37: **end if**
38: Update $\mathcal{Q}_{t+1}$ if needed, such that $|\mathcal{Q}_{t+1}| = |V(P_{t+1})| - 5|\mathcal{L}_{t+1}|$.   ▷ update permissible vertices

---

are only created when the blue edges of magenta vertices are uncoloured as a side effect of path extensions and augmentations (see lines (8) and (25) of `DegreeGreedy`).

For each $t \geq 0$, define the random variables $X(t) := |V(P_t)|$, $B(t) := |\mathcal{B}_t|$, $R(t) := |\mathcal{R}_t|$, $M(t) := |\mathcal{M}_t|$, $L(t) := |\mathcal{L}_t| = B(t) + R(t) + M(t)$, and $Y(t) := |V(\mathcal{Y}_t)|$. For each $q \geq 0$ define $D_q(t)$ to be the number of unsaturated vertices adjacent to precisely q blue edges. We define the stopping time τ_q to be the smallest $t \geq 0$ such that $D_j(t) = 0$ for all $j < q$, and $D_q(t) > 0$. It is easy to check that τ_q is well-defined and is non-decreasing in q. By definition, $\tau_0 = 0$. Let us refer to *phase q* as those t such that $\tau_{q-1} \leq t < \tau_q$. Observe that during phase q, each unsaturated vertex (i.e., vertex of $[n] \setminus V(P_t)$) has blue degree $q - 1$ or q.

2.6. Analyzing phase q.
Suppose that $\tau_{q-1} \leq t < \tau_q$. It will be convenient to denote $D(t) := D_{q-1}(t)$. Given $k_1, k_2 \geq 0$, we say that $y \in [n] \setminus V(P_t)$ is of *type* (k_1, k_2), provided it is adjacent to k_1 blue edges within $\mathcal{B}_t$ and k_2 blue edges within $\mathcal{M}_t$. Similarly, $x \in \mathcal{B}_t \cup \mathcal{M}_t$ is of type (k_1, k_2), provided its (unique) *blue* edge connects to a vertex of type (k_1, k_2). We denote the number of unsaturated vertices of type (k_1, k_2) by $C_{k_1,k_2}(t)$, the blue vertices of type (k_1, k_2) by $B_{k_1,k_2}(t)$, and the magenta (mixed) vertices of type (k_1, k_2) by $M_{k_1,k_2}(t)$. Observe that $B_{k_1,k_2}(t) = k_1 \cdot C_{k_1,k_2}(t)$ and $M_{k_1,k_2}(t) = k_2 \cdot C_{k_1,k_2}(t)$. Moreover, $D_j(t) = \sum_{\substack{k_1,k_2: \\ k_1+k_2=j}} C_{k_1,k_2}(t)$.

In Subsection 2.8, we inductively define the functions x, r, y and c_{k_1,k_2} for $k_1 + k_2 \geq 0$, as well as a constant $\sigma_q \geq 0$, such that the following lemma holds:

Lemma 2.7. *A.a.s. $\tau_q \sim \sigma_q n$ for every $0 \leq q \leq N$.[2] Moreover, at step τ_q, a.a.s.*
$$X(\tau_q) \sim x(\sigma_q)n, \quad R(\tau_q) \sim r(\sigma_q)n, \quad Y(\tau_q) \sim y(\sigma_q)n,$$
$$C_{k_1,k_2}(\tau_q) \sim c_{k_1,k_2}(\sigma_q)n \quad \text{for all } (k_1, k_2) \text{ where } k_1 + k_2 = q.$$

Although the method in the proof of Lemma 2.7 is similar to that of Lemmas 2.1, 2.2, 2.3 and Proposition 2.4, the analysis is much more intricate and involved. We postpone the proof until afterwards, and first complete the proof of Theorem 1.1.

2.7. Proving Theorem 1.1 assuming Lemma 2.7.

Proof of Theorem 1.1. Set $N = 100$. By Lemma 2.7, the execution of `DegreeGreedy` ends at some step $t_0 \sim \sigma_N n$. Moreover, $X(t_0) \sim x(\sigma_N)n$, $Y(t_0) \sim y(\sigma_N)n$, $R(t_0) \sim r(\sigma_N)n$ and $M(t_0) \sim m(\sigma_N)n$. Numerical computations show that $\sigma_N \approx 1.80249$. Let P be the path constructed after the first t_0 rounds, $\mathcal{Y}$ be the edges of $\mathcal{Y}_{t_0}$, and $\mathcal{E}$ be the red edges adjacent to the vertices of $\mathcal{M}_{t_0} \cup \mathcal{R}_{t_0}$. By the definition of `DegreeGreedy`, in particular by the way that $\mathcal{Y}_t$ is extended, and the way that the red edges are created, $\mathcal{Y}$ has the uniform distribution over all possible $|\mathcal{Y}|$ pairs over vertices that are not on the path P; and for each red edge, its end that is not on the path P is also uniformly distributed. Thus, $(P, \mathcal{Y}, \mathcal{E})$ has the distribution required for the analysis of `FullyRandomized`. Let

$$\hat{x} := x(\sigma_N) \approx 0.99991$$
$$\hat{y} := y(\sigma_N) \approx 0.0000029724$$
$$\hat{\ell}_1 := m(\sigma_N) + r(\sigma_N) \approx 0.00019429.$$

Then, $\hat{x}$, $\hat{y}$, and $\hat{\ell}_1$ satisfy $|P| \sim \hat{x}n$, $|\mathcal{Y}| \sim \hat{y}n$ and $|\mathcal{E}| \sim \hat{\ell}_1 n$. (The final equation holds since each vertex of $\mathcal{M}_{t_0} \cup \mathcal{R}_{t_0}$ is adjacent to one red edge.)

We next execute `FullyRandomized` with initial input $(P, \mathcal{Y}, \mathcal{E})$. Let α^* be as defined in (10) where the initial conditions to the differential equations (6)–(9) are set by $x(0) = \hat{x}$, $\ell_1(0) = \hat{\ell}_1$ and $\ell_2(0) = 0$. Numerical computations show that $\alpha^* \approx 0.014468$. By Proposition 2.4 and the fact that $\alpha^* < 3$, the execution of the first two stages (`DegreeGreedy` and `FullyRandomized`) finishes

---

[2]For functions $f = f(n)$ and $g = g(n)$, $f \sim g$ is shorthand for $f = (1 + o(1))g$.

at some step $(\sigma_N + \alpha^* + o(1))n \leq 1.81696n$, and the number of unsaturated vertices remaining is $o(n)$. Finally, the clean-up algorithm constructs a Hamiltonian cycle with an additional $o(n)$ steps by Lemma 2.5. The theorem follows. $\qquad\square$

2.8. **Proving Lemma 2.7.** We once again must first argue that our random variables cannot change drastically in one round during phase q.

Lemma 2.8 (Lipschitz Condition – `DegreeGreedy`)**.** *If* $|\Delta C(t)| := \max_{\substack{k_1,k_2 \in \mathbb{N} \cup \{0\}: \\ k_1 + k_2 \in \{q-1, q\}}} |\Delta C_{k_1,k_2}(t)|$,

then with probability $1 - O(n^{-1})$,

$$\max\{|\Delta X(t)|, |\Delta C(t)|, |\Delta R(t)|, |\Delta Y(t)|\} = O(\log n)$$

for all $\tau_{q-1} \leq t < \tau_q$ *with* $n - X(t) = \Omega(n)$.

Proof. Since $q \leq N$ is a constant which does not depend on n, we can apply the same argument to bound the red edges of each $\Delta C_{k_1,k_2}(t)$ as in Lemma 2.1, and then union bound over all $k_1, k_2 \geq 0$ such that $k_1 + k_2 \in \{q-1, q\}$. $\qquad\square$

Let H_t denote the history of the above random variables during the first t rounds. We now state the conditional expected differences of our random variables, where we assume that $\tau_{q-1} \leq t < \tau_q$ is such that $n - X(t) = \Omega(n)$. It will be convenient to define auxiliary random variables $A(t) := 2Y(t)/(1 - X(t))$ and $\Gamma(t) := 1 + Y(t)/(1 - X(t))$. Then,

$$\mathbb{E}[\Delta X(t) \mid H_t] = \frac{2Y(t)}{n} + \frac{2L(t)}{n}\,\Gamma(t) + O(1/n) \tag{11}$$

and

$$\mathbb{E}[\Delta Y(t) \mid H_t] = -\frac{2Y(t)}{n} + 2\left(1 - \frac{X(t) - Y(t)}{n}\right) - \frac{2L(t)}{n}\,A(t). \tag{12}$$

We omit the proofs of (11) and (12), as the derivation is the same as the analogous equations of Lemma 2.2. For the remaining random variables, we state the expected differences and derive them afterwards.

First, consider $\Delta R(t)$:

$$
\begin{aligned}
\mathbb{E}[\Delta R(t) \mid H_t] = {} & \frac{Y(t)}{n}\left(\frac{2M(t)}{n - X(t)} - \frac{2R(t)}{n - X(t)}\right) - \frac{2(B(t) + M(t))}{n}\frac{R(t)\Gamma(t)}{n - X(t)} \\
& + \sum_{\substack{j,h: \\ j+h \in \{q-1,q\}}} \frac{2B_{j,h}(t)}{n}\left(h + \frac{Y(t)}{n - X(t)}\frac{M(t)}{n - X(t)}\right) \\
& + \sum_{\substack{j,h: \\ j+h \in \{q-1,q\}}} \frac{2M_{j,h}(t)}{n}\left(h + \frac{Y(t)}{n - X(t)}\frac{M(t)}{n - X(t)}\right) \\
& - \frac{2R(t)}{n}\left(1 + \frac{R(t)\Gamma(t)}{n - X(t)}\right) + \frac{2R(t)}{n}\frac{M(t)\Gamma(t)}{n - X(t)} - \frac{R(t)}{n} + O(1/n).
\end{aligned} \tag{13}
$$

If $k_1 + k_2 = q - 1$, then $\Delta C_{k_1,k_2}(t)$ satisfies:

$$
\begin{aligned}
\mathbb{E}[\Delta C_{k_1,k_2}(t) \mid H_t] = {} & \frac{Y(t)}{n}\left(\frac{2M_{k_1-1,k_2+1}(t)}{n - X(t)} \cdot \mathbf{1}_{k_1 > 0} - \frac{2C_{k_1,k_2}(t)}{n - X(t)} - \frac{2M_{k_1,k_2}(t)}{n - X(t)}\right) \\
& + \frac{2(B(t) + M(t))}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n - X(t)} \cdot \mathbf{1}_{k_1 > 0} - \frac{M_{k_1,k_2}(t)}{n - X(t)}\right)\Gamma(t) \\
& - \frac{2(B(t) + M(t))}{n}\frac{A(t)}{2}\frac{C_{k_1,k_2}(t)}{n - X(t)} - \frac{2B_{k_1,k_2}(t)}{n} - \frac{2M_{k_1,k_2}(t)}{n}
\end{aligned}
$$

$$+ \frac{2R(t)}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)} - \frac{C_{k_1,k_2}(t)}{n-X(t)} \right) \Gamma(t)$$

$$- \frac{(X(t)-5L(t))}{n}\frac{C_{k_1,k_2}(t)}{D(t)} - \frac{R(t)}{n}\frac{C_{k_1,k_2}(t)}{D(t)}$$

$$+ \frac{B_{k_1+1,k_2-1}(t)}{n} \cdot \mathbf{1}_{k_2>0} - \frac{B_{k_1,k_2}(t)}{n} + O(1/n). \tag{14}$$

When $k_1 + k_2 = q$, two terms from the above expression are modified slightly, and have their signs reversed:

$$\mathbb{E}[\Delta C_{k_1,k_2}(t) \mid H_t] = \frac{Y(t)}{n}\left(\frac{2M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{2C_{k_1,k_2}(t)}{n-X(t)} - \frac{2M_{k_1,k_2}(t)}{n-X(t)} \right)$$

$$+ \frac{2(B(t)+M(t))}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)} \right)\Gamma(t)$$

$$- \frac{2(B(t)+M(t))}{n}\frac{A(t)}{2}\frac{C_{k_1,k_2}(t)}{n-X(t)} - \frac{2B_{k_1,k_2}(t)}{n} - \frac{2M_{k_1,k_2}(t)}{n}$$

$$+ \frac{2R(t)}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)} - \frac{C_{k_1,k_2}(t)}{n-X(t)} \right)\Gamma(t)$$

$$+ \frac{(X(t)-5L(t))}{n}\frac{C_{k_1-1,k_2}(t)}{D(t)} + \frac{R(t)}{n}\frac{C_{k_1,k_2-1}(t)}{D(t)}$$

$$+ \frac{B_{k_1+1,k_2-1}(t)}{n} \cdot \mathbf{1}_{k_2>0} - \frac{B_{k_1,k_2}(t)}{n} + O(1/n). \tag{15}$$

In order to prove the expected differences, we analyze the expected values of the random variables $\Delta R(t)$ and $\Delta C_{k_1,k_2}(t)$ when u_{t+1} lands in a subset $A \subseteq [n]$ for a number of choices of A. More precisely, we derive tables for $\mathbb{E}[\Delta R(t) \cdot \mathbf{1}_{u_{t+1} \in A} \mid H_t]$ and $\mathbb{E}[\Delta C_{k_1,k_2}(t) \cdot \mathbf{1}_{u_{t+1} \in A} \mid H_t]$ when $A \subseteq [n]$ varies across a number of subsets. Since these are disjoint subsets of $[n]$, and the random variables are 0 if u_{t+1} lands outside of these subsets, we can sum the second column entries to get the claimed expected differences. Note that the entries of our tables do *not* contain the often necessary $O(1/n)$ term.

In all our below explanations, we abuse notation and simultaneously identify our random variables as sets (i.e., $C_{k_1,k_2}(t)$ denotes the set of unsaturated vertices of type (k_1, k_2) after t steps).

$A \subseteq [n]$	$\mathbb{E}[\Delta R(t) \cdot \mathbf{1}_{u_{t+1} \in A} \mid H_t]$
U_t	0
$V(\mathcal{Y}_t)$	$\frac{Y(t)}{n}\left(\frac{2M(t)}{n-X(t)} - \frac{2R(t)}{n-X(t)} \right)$
Path distance 1 from $\mathcal{B}_t$	$-\frac{2B(t)}{n}\frac{R(t)\Gamma(t)}{n-X(t)} + \sum\limits_{\substack{j,h: \\ j+h \in \{q-1,q\}}} \frac{2B_{j,h}(t)}{n}\left(h + \frac{Y(t)}{n-X(t)}\frac{M(t)}{n-X(t)} \right)$
Path distance 1 from $\mathcal{M}_t$	$-\frac{2M(t)}{n}\frac{R(t)\Gamma(t)}{n-X(t)} + \sum\limits_{\substack{j,h: \\ j+h \in \{q-1,q\}}} \frac{2M_{j,h}(t)}{n}\left(h + \frac{Y(t)}{n-X(t)}\frac{M(t)}{n-X(t)} \right)$
Path distance 1 from $\mathcal{R}_t$	$-\frac{2R(t)}{n}\left(1 + \frac{R(t)\Gamma(t)}{n-X(t)} \right) + \frac{2R(t)}{n}\frac{M(t)\Gamma(t)}{n-X(t)}$
$\mathcal{R}_t$	$\frac{-R(t)}{n}$

TABLE 1. Expected Changes to $\Delta R(t)$.

Proof Sketch of Table 1. We provide complete proofs only of row entries 2 and 3, as the remaining entries follow similarly.

In order to see the second entry, expose the blue edges adjacent to $\mathcal{M}_t$, and the red edges adjacent to $\mathcal{R}_t$. If we then fix an arbitrary edge $yy' \in \mathcal{Y}_t$, we know that it is distributed u.a.r. amongst $\binom{[n]\setminus V(P_t)}{2}$. Thus, in expectation there are $2M(t)/(n-X(t))$ blue edges adjacent to the vertices of yy'. Now, if u_{t+1} lands on y or y', then a path augmentation is made, and these blue edges are destroyed, thus converting $2M(t)/(n-X(t))$ blue vertices to red vertices in expectation. Since this event occurs with probability $Y(t)/n$, this accounts for the $\frac{Y(t)}{n}\frac{2M(t)}{n-X(t)}$ term. An analogous argument applies to the $\frac{-Y(t)}{n}\frac{2R(t)}{n-X(t)}$ term.

Consider now the third row entry, where we shall first derive the $-\frac{2B(t)}{n}\frac{R(t)}{n-X(t)}\left(1 + \frac{Y(t)}{n-X(t)}\right)$ term. Suppose that b is a blue vertex, with blue edge bx. Observe that there are $R(t)/(n-X(t))$ red edges adjacent to x in expectation. Thus, $\frac{R(t)}{n-X(t)}$ red edges are destroyed in expectation if u_{t+1} lands next to a blue vertex. Since u_{t+1} lands next to a blue vertex with probability $2B(t)/n$, the $-\frac{2B(t)}{n}\frac{R(t)}{n-X(t)}$ term follows. The $-\frac{2B(t)}{n}\frac{R(t)}{n-X(t)}\frac{Y(t)}{1-X(t)}$ term follows by observing that $x = y$ for some $yy' \in \mathcal{Y}_t$ with probability $\frac{Y(t)}{1-X(t)}$. When this occurs, by computing the expected number of red edges adjacent to y', an additional $\frac{2B(t)}{n}\frac{R(t)}{n-X(t)}$ red edges are destroyed in expectation. These two cases account for the $-\frac{2B(t)}{n}\frac{R(t)}{n-X(t)}\left(1 + \frac{Y(t)}{n-X(t)}\right)$ term.

We now derive the term:

$$\sum_{\substack{j,h: \\ j+h \in \{q-1,q\}}} \frac{2B_{j,h}(t)}{n}\left(h + \frac{Y(t)}{n-X(t)}\frac{M(t)}{n-X(t)}\right).$$

Suppose that u_{t+1} lands next to a blue vertex of $B_{j,h}(t)$, which occurs with probability $2B_{j,h}(t)/n$. Let b be such a vertex, and denote its blue edge by bx. Now, by definition, there are h magenta vertices whose blue edge is also incident to x. We claim that all h of these magenta vertices will be reclassified as red vertices provided the following event occurs:

- All the red edges of these h magenta vertices are *not* adjacent to x.

The latter occurs with probability $\left(1 - \frac{1}{n-X(t)}\right)^h = 1 - O(1/n)$, since h is a constant, and $n-X(t) = \Omega(n)$. By summing over $j,h \in \{q-1,q\}$, this yields the expression $\sum_{\substack{j,h: \\ j+h \in \{q-1,q\}}} \frac{2hB_{j,h}(t)}{n} + O\left(\frac{1}{n}\right)$, and the $\sum_{\substack{j,h: \\ j+h \in \{q-1,q\}}} \frac{2B_{j,h}(t)}{n}\frac{Y(t)}{n-X(t)}\frac{M(t)}{n-X(t)}$ term follows similarly. $\qquad\square$

Consider now $\Delta C_{k_1,k_2}(t)$, where $k_1 + k_2 = q - 1$.

$A \subseteq [n]$	$\mathbb{E}[\Delta C_{k_1,k_2}(t) \cdot \mathbf{1}_{u_{t+1} \in A} \mid H_t]$
U_t	0
$V(\mathcal{Y}_t)$	$\frac{Y(t)}{n} \left(\frac{2M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{2C_{k_1,k_2}(t)}{n-X(t)} - \frac{2M_{k_1,k_2}(t)}{n-X(t)} \right)$
Path distance 1 from $\mathcal{B}_t$	$\frac{2B(t)}{n} \left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)} \right) \Gamma(t) - \frac{2B(t)}{n} \frac{A(t)}{2} \frac{C_{k_1,k_2}(t)}{n-X(t)} - \frac{2B_{k_1,k_2}(t)}{n}$
Path distance 1 from $\mathcal{M}_t$	$\frac{2M(t)}{n} \left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)} \right) \Gamma(t) - \frac{2M(t)}{n} \frac{A(t)}{2} \frac{C_{k_1,k_2}(t)}{n-X(t)} - \frac{2M_{k_1,k_2}(t)}{n}$
Path distance 1 from $\mathcal{R}_t$	$\frac{2R(t)}{n} \left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)} - \frac{C_{k_1,k_2}(t)}{n-X(t)} \right) \Gamma(t)$
$\mathcal{Q}_t$	$-\frac{(X(t)-5L(t))}{n} \frac{C_{k_1,k_2}(t)}{D(t)}$
$\mathcal{R}_t$	$-\frac{R(t)}{n} \frac{C_{k_1,k_2}(t)}{D(t)}$
$\mathcal{B}_t$	$-\frac{B_{k_1,k_2}(t)}{n} + \frac{B_{k_1+1,k_2-1}(t)}{n} \cdot \mathbf{1}_{k_2>0}$

TABLE 2. Expected Changes to $\Delta C_{k_1,k_2}(t)$ for $k_1 + k_2 = q - 1$.

Proof Sketch of Table 2. Assume that k_1, k_2 are both non-zero, as this is the most involved case. We provide complete proofs of row entries $2, 5$ and 6.

We begin with row entry 2. Observe that u_{t+1} lands on a vertex of an edge of $\mathcal{Y}_t$, say yy', with probability $Y(t)/n$. At this point, a path augmentation occurs, and yy' is added to the current path P_t, thus destroying the red and blue edges adjacent to y and y'. We claim that

$$\mathbb{E}[\Delta C_{k_1,k_2}(t) \mid H_t, \{u_{t+1} \in V(\mathcal{Y}_t)\}] = \frac{2M_{k_1-1,k_2+1}(t)}{n - X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{2C_{k_1,k_2}(t)}{n - X(t)} - \frac{2M_{k_1,k_2}(t)}{n - X(t)}.$$

Let us focus on the $-\frac{2C_{k_1,k_2}(t)}{n-X(t)} - \frac{2M_{k_1,k_2}(t)}{n-X(t)}$ term. In order to see this, fix an unsaturated vertex c of type (k_1, k_2). We shall prove that c is *destroyed* (i.e., removed from $C_{k_1,k_2}(t)$) with probability $(k_2 + 1)/(n - X(t)) + O(1/n^2)$, conditional on H_t and $\{u_{t+1} \in V(\mathcal{Y}_t)\}$. By summing over all $c \in C_{k_1,k_2}(t)$ and using the fact that $k_2 \cdot C_{k_1,k_2}(t) = M_{k_1,k_2}(t)$, this yields the $-\frac{2C_{k_1,k_2}(t)}{n-X(t)} - \frac{2M_{k_1,k_2}(t)}{n-X(t)}$ term. The $\frac{2M_{k_1-1,k_2+1}(t)}{n-X(t)}$ term follows similarly, where reclassifying unsaturated vertices of type $(k_1 - 1, k_2 + 1)$ to type (k_1, k_2) causes $C_{k_1,k_2}(t)$ to increase.

Let us now prove the above claim regarding $c \in C_{k_1,k_2}(t)$, where we condition on H_t and $u_{t+1} = y$ for some $yy' \in \mathcal{Y}_t$ in the below explanations. Suppose that $m_1, \ldots, m_{k_2}$ are the magenta neighbours of c of type (k_1, k_2). By definition, cm_i is coloured blue, and each m_i also has a red edge $m_i x_i$ for $i = 1, \ldots, k_2$. Observe that if either y or y' is equal to c, then c will be added to the path (and thus destroyed). Similarly, if x_i is added to the path, then the edge $m_i x_i$ is no longer red. In particular, m_i is converted to a blue vertex, and so the type of c is reclassified as $(k_1 + 1, k_2 - 1)$. In either case, c is destroyed. Now, $x_1, \ldots, x_{k_2}$ are distributed u.a.r. and independently amongst $[n] \setminus V(P_t)$, and so the vertices $c, x_1, \ldots, x_{k_2}$ are distinct with probability $\prod_{i=1}^{k_2} \left(1 - \frac{i}{n-X(t)} \right) = 1 - O(1/n)$, where we have used the fact that k_2 is a constant and $n - X(t) = \Omega(n)$. Moreover, yy' is distributed u.a.r. and independently amongst $\binom{[n] \setminus V(P_t)}{2}$. Thus, conditional on the vertices $c, x_1, \ldots, x_{k_2}$ being distinct, c is destroyed with probability

$$\frac{2(k_2 + 1)}{n - X(t)} - \frac{\binom{k_2+1}{2}}{\binom{n-X(t)}{2}} = \frac{2(k_2 + 1)}{n - X(t)} - O(1/n^2).$$

As such, c is destroyed with the claimed probability of $2(k_2 + 1)/(n - X(t)) + O(1/n^2)$.

Consider row entry 6. We begin by deriving the expression:

$$\frac{2R(t)}{n}\left(1+\frac{Y(t)}{1-X(t)}\right)\left(-\frac{M_{k_1,k_2}(t)}{n-X(t)}-\frac{C_{k_1,k_2}(t)}{n-X(t)}\right). \tag{16}$$

First, condition on the event when u_{t+1} lands at path distance one from some (red) vertex $r \in \mathcal{R}_t$. This occurs with probability $2R(t)/n$. Let rx be the unique red edge of r, where $x \in [n] \setminus V(P_t)$. We also condition on the event that $x = y$ for some $yy' \in \mathcal{Y}_t$, which occurs with probability $Y(t)/(1-X(t))$. Observe that when these events occur, `DegreeGreedy` adds x and y' to the path P_t via a path augmentation, and thus destroys the red and blue edges adjacent to x and y'.

Fix a vertex $c \in C_{k_1,k_2}(t)$. Conditional on the above events, we claim that c is destroyed with probability $2(k_2 + 1)/(n - X(t)) + O(1/n^2)$. To see this, observe that xy' is distributed u.a.r. amongst $\binom{[n] \setminus V(P_t)}{2}$. Thus, the same argument used to derive row entry 2 applies in this case. By summing over all $c \in C_{k_1,k_2}(t)$ (and multiplying by $2R(t)/n$ and $Y(t)/(1-X(t))$), this yields the term

$$\frac{2R(t)}{n}\frac{Y(t)}{1-X(t)}\left(-\frac{2M_{k_1,k_2}(t)}{n-X(t)}-\frac{2C_{k_1,k_2}(t)}{n-X(t)}\right).$$

In order to complete the derivation of (16), we consider the event when x is *not* a vertex of an edge of $\mathcal{Y}_t$. In this case, we only need to account for the red and blue edges destroyed when x is added to the path. This yields the following term:

$$\frac{2R(t)}{n}\left(1-\frac{Y(t)}{1-X(t)}\right)\left(-\frac{M_{k_1,k_2}(t)}{n-X(t)}-\frac{C_{k_1,k_2}(t)}{n-X(t)}\right).$$

The remaining term $\frac{2R(t)}{n}\left(1+\frac{Y(t)}{1-X(t)}\right)\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)}\cdot\mathbf{1}_{k_1>0}$ missing from (16) follows by a similar argument, where we destroy vertices of $C_{k_1-1,k_2+1}(t)$ to create new vertices of $C_{k_1,k_2}(t)$.

Let us now consider row entry 6 when u_{t+1} lands on a permissible vertex $x \in \mathcal{Q}_t$. Clearly, this event occurs with probability $|\mathcal{Q}_t|/n = (X(t)-5L(t))/n$. In this case, the algorithm chooses v_{t+1} u.a.r. amongst $D(t)$, the unsaturated vertices of minimum degree $q-1$, and colours the edge xv_{t+1} blue. Thus, if we fix $c \in C_{k_1,k_2}(t)$, then c will be chosen with probability $1/D(t)$ since $k_1+k_2 = q-1$. In this case, c gains a blue edge connected to a blue vertex, and thus will be reclassified as type (k_1+1, k_2). Thus, each $c \in C_{k_1,k_2}(t)$ will be reclassified with probability $\frac{X(t)-5L(t)}{n}\frac{1}{D(t)}$. By summing over all $c \in C_{k_1,k_2}(t)$, we get the $-\frac{(X(t)-5L(t))}{n}\frac{C_{k_1,k_2}(t)}{D(t)}$ term. $\qquad\square$

Finally, when $k_1 + k_2 = q$, the expressions in rows 6 and 7 are modified slightly.

$A \subseteq [n]$	$\mathbb{E}[\Delta C_{k_1,k_2}(t) \cdot \mathbf{1}_{u_{t+1}\in A} \mid H_t]$
U_t	0
$V(\mathcal{Y}_t)$	$\frac{Y(t)}{n}\left(\frac{2M_{k_1-1,k_2+1}(t)}{n-X(t)}\cdot\mathbf{1}_{k_1>0} - \frac{2C_{k_1,k_2}(t)}{n-X(t)} - \frac{2M_{k_1,k_2}(t)}{n-X(t)}\right)$
Path distance 1 from $\mathcal{B}_t$	$\frac{2B(t)}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)}\cdot\mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)}\right)\Gamma(t) - \frac{2B(t)}{n}\frac{A(t)}{2}\frac{C_{k_1,k_2}(t)}{n-X(t)} - \frac{2B_{k_1,k_2}(t)}{n}$
Path distance 1 from $\mathcal{M}_t$	$\frac{2M(t)}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)}\cdot\mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)}\right)\Gamma(t) - \frac{2M(t)}{n}\frac{A(t)}{2}\frac{C_{k_1,k_2}(t)}{n-X(t)} - \frac{2M_{k_1,k_2}(t)}{n}$
Path distance 1 from $\mathcal{R}_t$	$\frac{2R(t)}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)}\cdot\mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)} - \frac{C_{k_1,k_2}(t)}{n-X(t)}\right)\Gamma(t)$
$\mathcal{Q}_t$	$\frac{(X(t)-5L(t))}{n}\frac{C_{k_1-1,k_2}(t)}{D(t)}$
$\mathcal{R}_t$	$\frac{R(t)}{n}\frac{C_{k_1,k_2-1}(t)}{D(t)}$
$\mathcal{B}_t$	$-\frac{B_{k_1,k_2}(t)}{n} + \frac{B_{k_1+1,k_2-1}(t)}{n}\cdot\mathbf{1}_{k_2>0}$

TABLE 3. Expected Changes to $\Delta C_{k_1,k_2}(t)$ for $k_1 + k_2 = q$

Proof Sketch of Table 3. The explanations for the case of $k_1 + k_2 = q$ are identical to those of $k_1 + k_2 = q - 1$, except that vertices of type (k_1, k_2) are created (instead of destroyed) when u_{t+1} satisfies $u_{t+1} \in \mathcal{Q}_t$ or $u_{t+1} \in \mathcal{R}_t$. $\qquad\square$

We are now ready to inductively prove Lemma 2.7. Firstly, when $q = 0$, by definition $\tau_0 = 0$, and so $\sigma_0 := 0$ trivially satisfies the conditions of Lemma 2.7. Let us now assume that $q \geq 1$ and for each of $0 \leq i \leq q-1$ we have defined σ_i and functions x, r, y and $c_{j,h}$ on $[0, \sigma_i]$ for each $j, h \geq 0$ with $j + h = i$, and Lemma 2.7 holds for all $0 \leq i \leq q-1$. We shall define σ_q which satisfies $\sigma_q > \sigma_{q-1}$, extend each x, r, y and $c_{j,h}$ to $[0, \sigma_q]$, and define new functions c_{k_1,k_2} on $[0, \sigma_q]$ for $k_1 + k_2 = q$. We shall then prove that these functions satisfy the assertion of Lemma 2.7 with respect to τ_q and σ_q, which will complete the proof of the lemma.

Fix a sufficiently small constant $\varepsilon > 0$, and define the bounded domain $\mathcal{D}_\varepsilon$ as the points $(s, x, y, r, (c_{j,h})_{j+h\in\{q-1,q\}})$ such that

$$\sigma_{q-1} - 1 < s < 3, |x| < 1 - \varepsilon, |y| < 2, |r| < 2, |c_{j,h}| < 2, \varepsilon < \sum_{j,h:\,j+h=q-1} c_{j,h} < 2.$$

It will be convenient to define auxiliary functions to simplify our equations below. Specifically, set $b_{k_1,k_2} = k_1 \cdot c_{k_1,k_2}$ and $m_{k_1,k_2} := k_2 \cdot c_{k_1,k_2}$, as well as $b = \sum_{\substack{j,h:\\j+h\in\{q-1,q\}}} b_{j,h}$ and $m = \sum_{\substack{j,h:\\j+h\in\{q-1,q\}}} m_{j,h}$. Finally, set $\ell = b + m + r$ and $d = \sum_{\substack{j,h:\\j+h=q-1}} c_{j,h}$, as well as $\gamma = 1 + y/(1-x)$ and $a = 2y/(1-x)$. Observe the following system of differential equations:

$$x' = 2(y + \ell\gamma) \tag{17}$$
$$y' = -2y + 2(1 - x - y) - 2a\ell, \tag{18}$$

and

$$
\begin{aligned}
r' = &\; y\left(\frac{2m}{1-x} - \frac{2r}{1-x}\right) - 2(b+m)\frac{r\gamma}{1-x} \\
&\; + \sum_{\substack{j,h:\\j+h\in\{q-1,q\}}} 2b_{j,h}\left(h + \frac{y}{1-x}\frac{m}{1-x}\right)
\end{aligned}
$$

$$+ \sum_{\substack{j,h: \\ j+h \in \{q-1,q\}}} 2m_{j,h}\left(h + \frac{y}{1-x}\frac{m}{1-x}\right)$$

$$-2r\left(1 + \frac{r\gamma}{1-x}\right) + 2r\frac{m\gamma}{1-x} - r + O(1/n). \tag{19}$$

If $k_1 + k_2 = q - 1$, then:

$$
\begin{aligned}
c'_{k_1,k_2} = \ & y\left(\frac{2m_{k_1-1,k_2+1}}{1-x}\cdot \mathbf{1}_{k_1>0} - \frac{2c_{k_1,k_2}}{1-x} - \frac{2m_{k_1,k_2}}{1-x}\right) \\
& + 2(b+m)\left(\frac{m_{k_1-1,k_2+1}}{1-x}\cdot \mathbf{1}_{k_1>0} - \frac{m_{k_1,k_2}}{1-x}\right)\gamma \\
& - 2(b+m)\frac{a}{2}\frac{c_{k_1,k_2}}{1-x} - 2b_{k_1,k_2} - 2m_{k_1,k_2} \\
& + 2r\left(\frac{m_{k_1-1,k_2+1}}{1-x}\cdot \mathbf{1}_{k_1>0} - \frac{m_{k_1,k_2}}{1-x} - \frac{c_{k_1,k_2}}{1-x}\right)\gamma \\
& - (x - 5\ell)\frac{c_{k_1,k_2}}{d} - r\frac{c_{k_1,k_2}}{d} + b_{k_1+1,k_2-1}\cdot \mathbf{1}_{k_2>0} - b_{k_1,k_2}.
\end{aligned}
\tag{20}
$$

Otherwise, that is, if $k_1 + k_2 = q$, then:

$$
\begin{aligned}
c'_{k_1,k_2} = \ & y\left(\frac{2m_{k_1-1,k_2+1}}{1-x}\cdot \mathbf{1}_{k_1>0} - \frac{2c_{k_1,k_2}}{1-x} - \frac{2m_{k_1,k_2}}{1-x}\right) \\
& + 2(b+m)\left(\frac{m_{k_1-1,k_2+1}}{1-x}\cdot \mathbf{1}_{k_1>0} - \frac{m_{k_1,k_2}}{1-x}\right)\gamma \\
& - 2(b+m)\frac{a}{2}\frac{c_{k_1,k_2}}{1-x} - 2b_{k_1,k_2} - 2m_{k_1,k_2} \\
& + 2r\left(\frac{m_{k_1-1,k_2+1}}{1-x}\cdot \mathbf{1}_{k_1>0} - \frac{m_{k_1,k_2}}{1-x} - \frac{c_{k_1,k_2}}{1-x}\right)\gamma \\
& + (x - 5\ell)\frac{c_{k_1-1,k_2}}{d} + r\frac{c_{k_1,k_2-1}}{d} + b_{k_1+1,k_2-1}\cdot \mathbf{1}_{k_2>0} - b_{k_1,k_2}.
\end{aligned}
\tag{21}
$$

The right-hand side of each of the above equations is Lipchitz on the domain $\mathcal{D}_\varepsilon$, as d is bounded below by ε, and $|x| < 1 - \varepsilon$. Define

$$T_{\mathcal{D}_\varepsilon} := \min\{t \geq 0 : (t/n, X(t)/n, Y(t)/n, R(t)/n, (C_{k_1,k_2}(t)/n)_{k_1+k_2\in\{q,q-1\}}) \notin \mathcal{D}_\varepsilon\}.$$

By the inductive assumption, the 'Initial Condition' of Theorem A.1 is satisfied for some $\lambda = o(1)$ and values $\sigma_{q-1}, x(\sigma_{q-1}), r(\sigma_{q-1})$ and $c_{j,h}(\sigma_{q-1})$, where $c_{j,h}(\sigma_{q-1}) := 0$ for $j + h = q$. Moreover, the 'Trend Hypothesis' is satisfied with $\delta = O(1/n)$, by the expected differences of (11)–(15). Finally, the 'Boundedness Hypothesis' is satisfied with $\beta = O(\log n)$ and $\beta' = O(n^{-1})$ by Lemma 2.8. Thus, by Theorem A.1, for every $\xi > 0$, a.a.s. $X(t) = nx(t/n) + o(n)$, $R(t) = nr(t/n) + o(n), Y(t) = ny(t/n) + o(n)$ and $C_{k_1,k_2}(t) = nc_{k_1,k_2}(t/n) + o(n)$ uniformly for all $\sigma_{q-1}n \leq t \leq (\sigma(\varepsilon) - \xi)n$, where x, r, y and c_{k_1,k_2} are the unique solution to (17)–(21) with the above initial conditions, and $\sigma(\varepsilon)$ is the supremum of s to which the solution can be extended before reaching the boundary of $\mathcal{D}_\varepsilon$. This immediately yields the following lemma.

Lemma 2.9 (Concentration of `DegreeGreedy`'s Random Variables). *For every $\xi > 0$, a.a.s. for all $\tau_{q-1} \leq t \leq (\sigma(\varepsilon) - \xi)n$ and $k_1, k_2 \geq 0$ such that $k_1 + k_2 \in \{q, q-1\}$,*

$$\max\{|X(t) - x(t/n)n|, |Y(t) - y(t/n)n|, |R(t) - r(t/n)n|, |C_{k_1,k_2}(t) - c_{k_1,k_2}(t/n)n|\} = o(n).$$

As $\mathcal{D}_\varepsilon \subseteq \mathcal{D}_{\varepsilon'}$ for every $\varepsilon > \varepsilon'$, $\sigma(\varepsilon)$ is monotonically nondecreasing as $\varepsilon \to 0$, and so $\sigma_q := \lim_{\varepsilon\to 0+}\sigma(\varepsilon)$ exists. Moreover, the derivatives of the functions x, r, y and c_{k_1,k_2} are uniformly

bounded on (σ_{q-1}, σ_q), as $d = \sum_{\substack{j,h:\\ j+h=q-1}} c_{j,h}$, so $c_{j,h}/d \leq 1$ for $j + h = q - 1$. This implies that the functions are uniformly continuous, and so (uniquely) continuously extendable to $[\sigma_{q-1}, \sigma_q]$. The following limits thus exist:

$$x(\sigma_q) \quad := \quad \lim_{s \to \sigma_q -} x(s) \tag{22}$$

$$y(\sigma_q) \quad := \quad \lim_{s \to \sigma_q -} y(s) \tag{23}$$

$$r(\sigma_q) \quad := \quad \lim_{s \to \sigma_q -} r(s) \tag{24}$$

$$c_{k_1,k_2}(\sigma_q) \quad := \quad \lim_{s \to \sigma_q -} c_{k_1,k_2}(s). \tag{25}$$

The random variables $|R(t)/n|, |Y(t)/n|$ and $|C_{k_1,k_2}(t)/n|$ for $k_1 + k_2 \in \{q, q-1\}$ are bounded by 1 for all t. Thus, when t/n approaches σ_q, $X(t)/n$ approaches 1, t/n approaches 3, or $D(t)/n := \sum_{\substack{j,h:\\ j+h=q-1}} C_{j,h}(t)/n$ approaches 0. Formally, we have the following proposition:

Proposition 2.10. *For every $\varepsilon > 0$, there exists $\xi > 0$ such that a.a.s. one of the following holds.*

- *$D(t) < \varepsilon n$ for all $t \geq (\sigma_q - \xi)n$;*
- *$X(t) > (1 - \varepsilon)n$ for all $t \geq (\sigma_q - \xi)n$;*
- *$\sigma_q = 3$.*

The ordinary differential equations (17)–(21) again do not have an analytical solution. However, numerical solutions show that $\sigma_q < 3$, and $x(\sigma_q) < 1$. Thus, after executing `DegreeGreedy` for $t = \sigma_q n + o(n)$ steps, there are $D(t) < \varepsilon n$ vertices of type $q - 1$ remaining for some $\varepsilon = o(1)$. At this point, by observing the numerical solution (22)–(25) at σ_q, we know that there exists some absolute constant $0 < p < 1$ such that $(X(t) - 5L(t))/n \geq p$, where we recall that $L(t)$ counts the total number of coloured vertices at time t. Hence, at each step, some vertex of type $q - 1$ becomes of type q with probability at least p. Thus, by applying Chernoff's bound, one can show that a.a.s. after another $O(\varepsilon n/p) = o(n)$ rounds, all vertices of type $q - 1$ are destroyed. It follows that a.a.s. $|\tau_q/n - \sigma_q| = o(1)$, and so Lemma 2.7 is proven.

3. Proof of Theorem 1.2

Suppose that G_t is the graph constructed by a strategy after t rounds, whose edges are $(u_i, v_i)_{i=1}^t$. When proving Theorem 1.2, it is convenient to refer to the *random vertex u_i as a* `square`, *and v_i as a* `circle` so that every edge in G_t joins a square with a circle. Recall that for a vertex $x \in [n]$, we say that the square u_i *lands* on x, or that x is *hit/covered* by u_i. We extend the analogous terminology to circles.

We begin with the following observations. Suppose that H is a Hamiltonian cycle created in the process. Then,

- (O_1) H uses exactly n squares;
- (O_2) H uses at most 2 squares on each vertex;
- (O_3) Suppose (u_i, v_i) is an edge of G_t, and v_i received at least two squares. Then, either H uses at most one square on v_i, or H does not contain the edge (u_i, v_i).

The first two observations above are straightforward. For (O_3), notice that if H uses exactly 2 squares on v_i, then these 2 squares correspond to 2 edges in H that are incident to v_i. Moreover, neither of these edges can be (u_i, v_i), as u_i is the square of (u_i, v_i). Thus, the edge (u_i, v_i) cannot be used by H as v_i has degree 2 in H.

Define Z_x as the number of squares on vertex $x \in [n]$. The observation (O_2) above indicates the consideration of the random variable

$$Z = \sum_{x=1}^{n} \left(\mathbf{1}_{Z_x=1} + 2 \cdot \mathbf{1}_{Z_x \geq 2}\right) = 2n - \sum_{x=1}^{n} \left(2 \cdot \mathbf{1}_{Z_x=0} + \mathbf{1}_{Z_x=1}\right),$$

which counts the total number of squares that can possibly contribute to H, truncated at 2 for each vertex. Observation (O_3) above indicates the consideration of the following two sets of structures. Let $\mathcal{W}_1$ be the set of pairs of vertices (x, y) at time t such that

  (a) x receives its first square at some step $i < t$, and y receives the corresponding circle in the same step;
  (b) no more squares land on x after step i;
  (c) at least two squares land on y after step i.

Let $\mathcal{W}_2$ be the set of pairs of vertices (x, y) at time t such that

  (a) x receives a square at some step $i < t$, and y receives the corresponding circle in the same step;
  (b) x receives exactly two squares (the other square may land on x either before or after step i);
  (c) at least two squares land on y after step i.

Note that for every $(x, y) \in \mathcal{W}_1$, at most 2 squares on x and y together can be used in H, although x and y together contribute 3 to the value of Z. Similarly, for every $(x, y) \in \mathcal{W}_2$, at most 3 squares on x and y together can be used in H, although x and y together contribute 4 to the value of Z. We prove the following upper bound on the total number of squares that can possibly contribute to the construction of H.

Let

$$\begin{aligned}
\mathcal{T}_1 &= \{((x_1, y_1), (x_2, y_2)) \in \mathcal{W}_1 \times \mathcal{W}_2 : \ y_1 = x_2\} \\
\mathcal{T}_2 &= \{((x_1, y_1), (x_2, y_2)) \in \mathcal{W}_2 \times \mathcal{W}_2 : \ y_1 = x_2\}.
\end{aligned}$$

Let $W := |\mathcal{T}_1| + |\mathcal{T}_2|$.

Claim 3.1. *The total number of squares contributing to H is at most $Z - |\mathcal{W}_1| - |\mathcal{W}_2| + W$.*

Proof of Claim 3.1. By the discussions above, the total number of squares contributing to H is at most $Z - |\mathcal{W}_1| - |\mathcal{W}_2| + Z'$ where Z' accounts for double counting caused by distinct $(x_1, y_1), (x_2, y_2) \in \mathcal{W}_1 \cup \mathcal{W}_2$ where $\{x_1, y_1\} \cap \{x_2, y_2\} \neq \emptyset$. We bound Z' by W by considering the following cases.

Case 1: We first consider the case that $(x_1, y_1), (x_2, y_2) \in \mathcal{W}_1$ where $(x_1, y_1) \neq (x_2, y_2)$ and $\{x_1, y_1\} \cap \{x_2, y_2\} \neq \emptyset$. The only possible situation is $y_1 = y_2$. In this case, only 2 squares out of the four squares on x_1, x_2 and y_1 that were counted by Z can contribute to H; hence there is no double counting (i.e. the offset is -2 as correctly carried out in $-|\mathcal{W}_1| - |\mathcal{W}_2|$).

Case 2: The second case involves $((x_1, y_1), (x_2, y_2)) \in \mathcal{W}_1 \times \mathcal{W}_2$ where $\{x_1, y_1\} \cap \{x_2, y_2\} \neq \emptyset$. By definition, $x_1 \neq x_2$ and $x_1 \neq y_2$. We consider the two possible subcases:

Case 2', $x_2 = y_1$: In this case, there are 5 squares on the three vertices x_1, $y_1 = x_2$, and y_2 that were counted by Z, and at most 4 can contribute to H by using the two squares on $y_1 = x_2$ and the two squares on y_2. Thus, each such structure causes one double counting and this explains the term $|\mathcal{T}_1|$ in W;

Case 2", $y_1 = y_2$: In this case, there are 5 squares on the three vertices x_1, x_2, and $y_1 = y_2$ that were counted by Z, and at most 3 can contribute to H. Thus, there is no double counting in this subcase.

Case 3: The third case involves $(x_1, y_1), (x_2, y_2) \in \mathcal{W}_2$ where $(x_1, y_1) \neq (x_2, y_2)$ and $\{x_1, y_1\} \cap \{x_2, y_2\} \neq \emptyset$. We consider the two possible subcases:

Case 3', $y_1 = x_2$ (and symmetrically $y_2 = x_1$): in this subcase, at most 5 squares out of the 6 squares on the three vertices that were counted by Z can be used to construct H (by using two squres on x_1; two squares on y_2 and one square on $y_1 = x_2$). Note also that for the two pairs of edges (x_1, y_1) and (x_2, y_2) considered here, we may always label them so that $y_1 = x_2$. This explains the $|\mathcal{T}_2|$ term in W.

Case 3", $y_1 = y_2$ or $x_1 = x_2$: at most 4 squares out of the 6 squares on the three vertices that were counted by Z can be used to construct H; thus there is no double counting in this case.   □

The random variable Z is well understood. From the limiting Poisson distribution of the number of squares in a single vertex, we immediately get that, a.a.s., $Z \sim (2 - 2e^{-s} - e^{-s}s)n$ for $s := t/n$.

We will estimate the expectation of $|\mathcal{W}_1|, |\mathcal{W}_2|, |\mathcal{T}_1|, |\mathcal{T}_2|$ as well as the concentration of these random variables. However, concentration may fail if the semi-random process uses a strategy which places many circles on a single vertex. Intuitively, placing many circles on a single vertex is not a good strategy for quickly building a Hamiltonian cycle, as it wastes many edges. To formalise this idea, let $\mu := \sqrt{n}$ (indeed, choosing any μ such that $\mu \to \infty$ and $\mu = o(n)$ will work). We say that a strategy for the semi-random process is *μ-well-behaved* up until step t, if no vertex receives more than μ circles in the first t steps. In [19, Definition 3.2 – Proposition 3.4], it was proven that it is sufficient to consider μ-well-behaved strategies in the first $t = O(n)$ steps for establishing a lower bound on the number of steps needed to build a perfect matching. These definitions and proofs can be easily adapted for building Hamilton cycles in an obvious way. We thus omit the details and only give a high-level explanation below.

The key idea is that within $t = O(n)$ steps of any semi-random process, the number of vertices that received more than μ circles is at most $O(n/\mu) = o(n)$. Therefore, if a Hamiltonian cycle C is built in t steps, then the subgraph H of C induced by the set S of vertices that received at most μ circles in G_t is a collection of paths spanning all vertices in S which must also contain $n - O(n/\mu) = (1 - o(1))n$ edges. We call such a pair (S, H) an *approximate Hamiltonian cycle*. It follows from the above argument that it takes at least as long time to build a Hamiltonian cycle as to build an approximate Hamiltonian cycle. It is then easy to show by a coupling argument that if a strategy builds an approximate Hamiltonian cycle in $t = O(n)$ steps, then there exists a well-behaved strategy that builds an approximate Hamiltonian cycle in t steps as well. Note that observations (O_2)–(O_3) hold for approximate Hamiltonian cycles, and (O_1) holds for approximate Hamiltonian cycles with n replaced by $(1 - o(1))n$. Thus, the following condition has to be satisfied at the time when an approximate Hamiltonian cycle is built:

$$Z - |\mathcal{W}_1| - |\mathcal{W}_2| + W \geq (1 - o(1))n.$$

We now estimate the sizes of $\mathcal{W}_1$, $\mathcal{W}_2$, $\mathcal{T}_1$, and $\mathcal{T}_2$ in the semi-random process when executing a well-behaved strategy $\mathcal{S}$. Crucially, the sizes of these sets do *not* rely on the decisions made by $\mathcal{S}$. Recall that $(G_s^{\mathcal{S}})_{s \geq 0}$ denotes the sequence of graphs produced by $\mathcal{S}$.

Lemma 3.2. *Suppose $\mathcal{S}$ is μ-well-behaved. For every $t = \Theta(n)$, a.a.s. the following holds in $G_t^{\mathcal{S}}$,*

$$Z - |\mathcal{W}_1| - |\mathcal{W}_2| + W \sim f(s)n,$$

where $s := t/n$ and $f(s)$ is defined as in Theorem 1.2.

Proof. Suppose $\mathcal{S}$ is μ-well-behaved until time $t = \Theta(n)$. We shall prove that a.a.s. the following properties hold for $G_t^{\mathcal{S}}$:

$$|\mathcal{W}_1| \sim n \sum_{i \leq t} \frac{1}{n} \left(1 - \frac{1}{n}\right)^t \sum_{i < j_1 < j_2 \leq t} \frac{1}{n^2} \left(1 - \frac{1}{n}\right)^{j_2} \tag{26}$$

$$|\mathcal{W}_2| \sim n \sum_{i_1 < i_2 \leq t} \frac{1}{n^2} \left(1 - \frac{1}{n}\right)^t \left(\sum_{i_1 < j_1 < j_2 \leq t} \frac{1}{n^2} \left(1 - \frac{1}{n}\right)^{j_2} + \sum_{i_2 < j_1 < j_2 \leq t} \frac{1}{n^2} \left(1 - \frac{1}{n}\right)^{j_2} \right) \tag{27}$$

$$|\mathcal{T}_1| \sim n \sum_{i \le t} \frac{1}{n}\left(1 - \frac{1}{n}\right)^t \sum_{i < j_1 < j_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^t$$

$$\times \left(\sum_{j_1 < h_1 < h_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^{h_2} + \sum_{j_2 < h_1 < h_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^{h_2}\right) \qquad (28)$$

$$|\mathcal{T}_2| \sim n \sum_{i_1 < i_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^t \sum_{i_1 < j_1 < j_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^t$$

$$\times \left(\sum_{j_1 < h_1 < h_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^{h_2} + \sum_{j_2 < h_1 < h_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^{h_2}\right)$$

$$+ \sum_{i_1 < i_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^t \sum_{i_2 < j_1 < j_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^t$$

$$\times \left(\sum_{j_1 < h_1 < h_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^{h_2} + \sum_{j_2 < h_1 < h_2 \le t} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^{h_2}\right). \qquad (29)$$

We prove (26) and briefly explain the expressions in (27)–(29) whose proofs are similar to that of (26). Fix a vertex $x \in [n]$ and a square u_i for $i \le t$. The probability that u_i lands on x in step i is $1/n$. Condition on this event. The probability that x receives no squares in any steps other than i is $(1 - 1/n)^{t-1} \sim (1 - 1/n)^t$. Let y be the vertex which the strategy chooses to pair u_i with. Fix any two integers $i < j_1 < j_2 \le t$, the probability that y receives its first two squares at times j_1 and j_2 is asymptotically $n^{-2}(1 - 1/n)^{j_2}$. Summing over all possible values of i, j_1, j_2 and multiplying by n, the number of choices for x, gives $\mathbb{E}|\mathcal{W}_1|$.

For concentration of $|\mathcal{W}_1|$ we prove that $\mathbb{E}|\mathcal{W}_1|^2 \sim (\mathbb{E}|\mathcal{W}_1|)^2$. For any pair of $((x_1, y_1), (x_2, y_2))$ in $\mathcal{W}_1 \times \mathcal{W}_1$, either x_1, y_1, x_2, y_2 are pairwise distinct, or $y_1 = y_2$. It is easy to see that the expected number of pairs where x_1, y_1, x_2, y_2 are pairwise distinct is asymptotically

$$n^2 \sum_{\substack{i_1 \le t \\ i_2 \le t}} \frac{1}{n^2}\left(1 - \frac{1}{n}\right)^{2t} \sum_{\substack{i_1 \le j_1 < j_2 \le t \\ i_2 \le h_1 < h_2 \le t}} \frac{1}{n^4}\left(1 - \frac{1}{n}\right)^{j_2 + h_2} \sim (\mathbb{E}|\mathcal{W}_1|)^2.$$

The expected number of pairs where $y_1 = y_2$ is at most μn as there are most n choices for x_1 and given (x_1, y_1), there can be at most μ choices for (x_2, y_2) since $\mathcal{S}$ is μ-well-behaved. Since $\mu = o(n)$, $\mu n = o(n^2)$ which is $o((\mathbb{E}|\mathcal{W}_1|)^2)$. Thus we have verified that $\mathbb{E}|\mathcal{W}_1|^2 \sim (\mathbb{E}|\mathcal{W}_1|)^2$ and thus by the second moment method, a.a.s. $|\mathcal{W}_1| \sim \mathbb{E}|\mathcal{W}_1|$.

The proofs for the expectation and concentration of $|\mathcal{W}_2|$, $|\mathcal{T}_1|$ and $|\mathcal{T}_2|$ are similar. We briefly explain the expressions in (27)–(29):

In (27), i_1 and i_2 denote the two steps at which x receives a square. Since there are two squares on x, there are two choices of circles, namely v_{i_1} and v_{i_2}. The two summations over (j_1, j_2) accounts for the two choices of v_{i_1} and v_{i_2}, depending on which is to be covered by two squares. Thus, j_1 and j_2 denote the steps where the first two squares on v_{i_1} or v_{i_2} arrive.

In (28), i denotes the step where x_1 receives its only square; j_1 and j_2 denote the two steps where $y_1 = x_2$ receives its two squares. Hence, there are two choices for y_2, and h_1 and h_2 denote the two steps of the first two squares y_2 receives.

In (29), i_1 and i_2 denote the two steps where x_1 receives its two squares—hence there are two choices for y_1. Integers j_1 and j_2 denote the two steps where $y_1 = x_1$ receives its two squares—hence there are two choices for y_2. Finally, h_1 and h_2 denote the steps where y_2 receives its first two squares.

By applying the above equations, we deduce that for $t = sn$,

$$|\mathcal{W}_1| \sim ne^{-s} \int_0^s dx \int_x^s dy_1 \int_{y_1}^s e^{-y_2} dy_2 = ne^{-s} \left(1 - \frac{e^{-s}s^2}{2} - e^{-s}s - e^{-s} \right)$$

$$|\mathcal{W}_2| \sim ne^{-s} \int_0^s dx_1 \int_{x_1}^s dx_2 \left(\int_{x_1}^s dy_1 \int_{y_1}^s e^{-y_2} dy_2 + \int_{x_2}^s dy_1 \int_{y_1}^s e^{-y_2} dy_2 \right)$$

$$= ne^{-s} \left(s - e^{-s}s^2 - \frac{e^{-s}s^3}{2} - e^{-s}s \right)$$

$$|\mathcal{T}_1| \sim ne^{-2s} \int_0^s dx \int_x^s dy_1 \int_{y_1}^s dy_2 \left(\int_{y_1}^s dz_1 \int_{z_1}^s e^{-z_2} dz_2 + \int_{y_2}^s dz_1 \int_{z_1}^s e^{-z_2} dz_2 \right)$$

$$= ne^{-2s} \left(-1 + s - \frac{e^{-s}s^3}{3} - \frac{e^{-s}s^2}{2} - \frac{e^{-s}s^4}{8} + e^s \right)$$

$$|\mathcal{T}_2| \sim ne^{-2s} \int_0^s dx_1 \int_{x_1}^s dx_2 \int_{x_1}^s dy_1 \int_{y_1}^s dy_2 \left(\int_{y_1}^s dz_1 \int_{z_1}^s e^{-z_2} dz_2 + \int_{y_2}^s dz_1 \int_{z_1}^t e^{-z_2} dz_2 \right)$$

$$+ ne^{-2s} \int_0^s dx_1 \int_{x_1}^s dx_2 \int_{x_2}^s dy_1 \int_{y_1}^s dy_2 \left(\int_{y_1}^s dz_1 \int_{z_1}^s e^{-z_2} dz_2 + \int_{y_2}^s dz_1 \int_{z_1}^s e^{-z_2} dz_2 \right)$$

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

It follows now that $Z - |\mathcal{W}_1| - |\mathcal{W}_2| + W \sim f(s)n$ where we recall that

$$f(s) = 2 + e^{-3s}(s + 1) \left(1 - \frac{s^2}{2} - \frac{s^3}{3} - \frac{s^4}{8} \right) + e^{-2s} \left(2s + \frac{5s^2}{2} + \frac{s^3}{2} \right) - e^{-s} \left(3 + 2s \right).$$

This finishes the proof of the lemma.   □

Proof of Theorem 1.2. Recall that β is the positive root of $f(s) = 1$. Then, for every $\varepsilon > 0$, $Z - |\mathcal{W}_1| - |\mathcal{W}_2| + W \leq (1 - O(\varepsilon))n$ a.a.s. in $G^{\mathcal{S}}_{(\beta-\varepsilon)n}$ for any μ-well-behaved $\mathcal{S}$. Therefore, $C_{\mathtt{HAM}} \geq \beta$.   □

4. CONCLUSION AND OPEN PROBLEMS

We have made significant progress on reducing the gap between the previous best upper and lower bounds on $C_{\mathtt{HAM}}$. That being said, we do not believe that any of our new bounds are tight. For instance, in the case of our lower bound, one could study the appearance of more complicated substructures which prevent any strategy from building a Hamiltonian cycle. One way to likely improve the upper bound would be to analyze an adaptive algorithm whose decisions are all made greedily. In the terminology of `DegreeGreedy`, when a (second) square lands on a blue vertex, choose the edge greedily chosen amongst unsaturated vertices of minimum blue degree (as opposed to u.a.r.). Unfortunately, it seems challenging to analyze this algorithm via the differential equation method.

Another direction is to understand which graph properties exhibit *sharp thresholds.* Given property $\mathcal{P}$, the definition of $C_{\mathcal{P}}$ ensures that there exists a strategy $\mathcal{S}^*$ such that for all $\varepsilon > 0$, $G^{\mathcal{S}^*}_t(n)$ satisfies $\mathcal{P}$ a.a.s. for $t \geq (C_{\mathcal{P}} + \varepsilon)n$. On the other hand, $G^{\mathcal{S}^*}_t(n)$ may satisfy $\mathcal{P}$ with *constant* probability for $t \leq (C_{\mathcal{P}} - \varepsilon)n$ without contradicting the definition of $C_{\mathcal{P}}$. For $\mathcal{P}$ to have a sharp threshold means that, for every strategy $\mathcal{S}$ and $\varepsilon > 0$, if $t \leq (C_{\mathcal{P}} - \varepsilon)n$ then, a.a.s., $G^{\mathcal{S}}_t(n)$ does *not* satisfy $\mathcal{P}$. It is known that for basic properties, such as minimum degree $k \geq 1$, sharp thresholds do exist [5]. Moreover, in [4] it was shown that if H is a spanning graph with maximum degree $\Delta = \omega(\log n)$, then the appearance of H takes $(\Delta/2 + o(\Delta))n$ rounds, and H (deterministically) cannot be constructed in fewer than $\Delta n/2$ rounds. However, in general it remains open as to whether or not a

sharp threshold exists when H is *sparse* (i.e., $\Delta = O(\log n)$). Recently, the third author and Surya developed a general machinery for proving the existence of sharp thresholds in adaptive random graph processes [24]. Applied to the semi-random graph process, they show that sharp thresholds exist for the property of being Hamiltonian and the property of containing a perfect matching. This provides some evidence that sharp thresholds *do* exist when $\Delta = O(\log n)$, and we leave this as an interesting open problem.

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. *arXiv preprint arXiv:2409.19335*, 2024.

[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] Patrick Bennett and Andrzej Dudek. A gentle introduction to the differential equation method and dynamic concentration. *Discrete Mathematics*, 345(12):113071, 2022.

[7] Patrick Bennett and Calum MacRury. Extending wormald's differential equation method to one-sided bounds. *Combinatorics, Probability and Computing*, in press, 2024.

[8] J. Bezanson, A. Edelman, S. Karpinski, and V.B. Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[9] Tom Bohman and Alan Frieze. Avoiding a giant component. *Random Structures & Algorithms*, 19(1):75–85, 2001.

[10] Tom Bohman and Alan Frieze. Hamilton cycles in 3-out. *Random Structures & Algorithms*, 35(4):393–417, 2009.

[11] Sofiya Burova and Lyuben Lichev. The semi-random tree process. *arXiv preprint arXiv:2204.07376*, 2022.

[12] Petruţ Cobârzan. A new upper bound for achlioptas processes. 10 2014.

[13] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 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 and Gregory B Sorkin. Hamilton cycles in a semi-random graph model. *arXiv preprint arXiv:2208.00255*, 2022.

[16] David Gamarnik, Mihyun Kang, and Pawel Pralat. Cliques, chromatic number, and independent sets in the semi-random process. *SIAM Journal on Discrete Mathematics*, 38(3):2312–2334, 2024.

[17] Pu Gao, Bogumił Kamiński, Calum MacRury, and Paweł Prałat. Hamilton cycles in the semi-random graph process. *European Journal of Combinatorics*, 99:103423, 2022.

[18] Pu Gao, Calum MacRury, and Pawel 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.

[19] Pu Gao, Calum MacRury, and Paweł Prałat. Perfect matchings in the semirandom graph process. *SIAM Journal on Discrete Mathematics*, 36(2):1274–1290, 2022.

[20] Shoni Gilboa and Dan Hefetz. Semi-random process without replacement. In *Extended Abstracts EuroComb 2021*, pages 129–135. Springer, 2021.

[21] Michal 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.

[22] Hidde Koerts. k-connectedness and k-factors in the semi-random graph process. Master's thesis, University of Waterloo, 2022.

[23] Michael Krivelevich, Eyal Lubetzky, and Benny Sudakov. Hamiltonicity thresholds in achlioptas processes. *Random Structures & Algorithms*, 37(1):1–24, 2010.

[24] Calum MacRury and Erlang Surya. Sharp thresholds in adaptive random graph processes. *Random Structures & Algorithms*, 64(3):741–767, 2024.

[25] Michael Molloy, Paweł Prałat, and Gregory B Sorkin. Perfect matchings and loose hamilton cycles in the semi-random hypergraph model. *Random Structures and Algorithms*, in print, 2024.

[26] Paweł Prałat and Harjas Singh. Power of k choices in the semi-random graph process. *Electronic Journal of Combinatorics*, 31(1):#P1.11, 2024.

[27] Joel Spencer and Nicholas Wormald. Birth control for giants. *Combinatorica*, 27, 07 2004.

[28] David W. Walkup. Matchings in random regular bipartite digraphs. *Discrete Mathematics*, 31(1):59–64, 1980.

[29] Lutz Warnke. On wormald's differential equation method. *arXiv preprint arXiv:1905.08928*, 2019.

[30] Nicholas C Wormald. The differential equation method for random graph processes and greedy algorithms. *Lectures on approximation and randomized algorithms*, 73:155, 1999.

Appendix A. The Differential Equation Method

In this section, we provide a self-contained *non-asymptotic* statement of the differential equation method. The statement combines [29, Theorem 2], and its extension [29, Lemma 9], in a form convenient for our purposes, where we modify the notation of [29] slightly. In particular, we rewrite [29, Lemma 9] in a less general form in terms of a stopping time T. We need only check the 'Boundedness Hypothesis' (see below) for $0 \le t \le T$, which is exactly the setting of Lemmas 2.1 and 2.8. A similar theorem is stated in [7, Theorem 2].

Suppose we are given integers $a, n \ge 1$, a bounded domain $\mathcal{D} \subseteq \mathbb{R}^{a+1}$, and functions $(F_k)_{1 \le k \le a}$ where each $F_k : \mathcal{D} \to \mathbb{R}$ is L-Lipschitz-continuous on $\mathcal{D}$ for $L \ge 0$. Moreover, suppose that $R \in [1, \infty)$ and $S \in (0, \infty)$ are *any* constants which satisfy $\max_{1 \le k \le a} |F_k(x)| \le R$ for all $x = (s, y_1, \ldots, y_a) \in \mathcal{D}$ and $0 \le s \le S$.

Theorem A.1 (Differential Equation Method, [29]). *Suppose we are given σ-fields $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots$, and for each $t \ge 0$, random variables $((Y_k(t))_{1 \le k \le a}$ which are $\mathcal{F}_t$-measurable. Define $T_{\mathcal{D}}$ to be the minimum $t \ge 0$ such that*

$$(t/n, Y_1(t)/n, \ldots, Y_k(t)/n) \notin \mathcal{D}.$$

Let $T \ge 0$ be an (arbitrary) stopping time[3] adapted to $(\mathcal{F}_t)_{t \ge 0}$, and assume that the following conditions hold for $\delta, \beta, \gamma \ge 0$ and $\lambda \ge \delta \min\{S, L^{-1}\} + R/n$:

(i) The 'Initial Condition': For some $(0, \hat{y}_1, \ldots, \hat{y}_a) \in \mathcal{D}$,

$$\max_{1 \le k \le a} |Y_k(0) - \hat{y}_k n| \le \lambda n.$$

(ii) The 'Trend Hypothesis': For each $t \le \min\{T, T_{\mathcal{D}} - 1\}$,

$$|\mathbb{E}[Y_k(t+1) - Y_k(t) \mid \mathcal{F}_t] - F_k(t/n, Y_1(t)/n, \ldots, Y_a(t)/n)| \le \delta.$$

(iii) The 'Boundedness Hypothesis': With probability $1 - \gamma$,

$$|Y_k(t+1) - Y_k(t)| \le \beta,$$

for each $t \le \min\{T, T_{\mathcal{D}} - 1\}$.

Then, with probability at least $1 - 2a \exp\left(\frac{-n\lambda^2}{8S\beta^2}\right) - \gamma$, we have that

$$\max_{0 \le t \le \min\{T, \sigma n\}} \max_{1 \le k \le a} |Y_k(t) - y_k(t/n)n| < 3\lambda \exp(LS)n, \tag{30}$$

where $(y_k(s))_{1 \le k \le a}$ is the unique solution to the system of differential equations

$$y_k'(s) = F_k(s, y_1(s), \ldots, y_a(s)) \quad \text{with } y_k(0) = \hat{y}_k \text{ for } 1 \le k \le a, \tag{31}$$

and $\sigma = \sigma(\hat{y}_1, \ldots, \hat{y}_a) \in [0, S]$ is any choice of $\sigma \ge 0$ with the property that $(s, y_1(s), \ldots, y_a(s))$ has ℓ^∞-distance at least $3\lambda \exp(LS)$ from the boundary of $\mathcal{D}$ for all $s \in [0, \sigma)$.

---

[3]The stopping time $T \ge 0$ is *adapted* to $(\mathcal{F}_t)_{t \ge 0}$, provided the event $\{\tau = t\}$ is $\mathcal{F}_t$-measurable for each $t \ge 0$.

Remark 3. Standard results for differential equations guarantee that (31) has a unique solution $(y_k(s))_{1 \le k \le a}$ which extends arbitrarily close to the boundary of $\mathcal{D}$.

DEPARTMENT OF MATHEMATICAL SCIENCES, CARNEGIE MELLON UNIVERSITY, PITTSBURGH PA 15213, USA
Email address: `alan@random.math.cmu.edu`

DEPARTMENT OF COMBINATORICS AND OPTIMIZATION, UNIVERSITY OF WATERLOO, WATERLOO, CANADA
Email address: `pu.gao@uwaterloo.ca`

GRADUATE SCHOOL OF BUSINESS, COLUMBIA UNIVERSITY, NEW YORK CITY, USA
Email address: `cm4379@columbia.edu`

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

DEPARTMENT OF MATHEMATICS, THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE, HOUGHTON STREET, LONDON WC2A 2AE, ENGLAND
Email address: `g.b.sorkin@lse.ac.uk`