A Fully Adaptive Strategy for Hamiltonian Cycles in the Semi-Random Graph Process

Pu Gao
Department of Combinatorics and Optimization, University of Waterloo, Canada

Calum MacRury
Department of Computer Science, University of Toronto, Canada

Paweł Prałat
Department of Mathematics, Toronto Metropolitan University, Canada

Abstract

The semi-random graph process is a single player game in which the player is initially presented an empty graph on n vertices. In each round, a vertex u is presented to the player independently and uniformly at random. The player then adaptively selects a vertex v, and adds the edge uv to the graph. For a fixed monotone graph property, the objective of the player is to force the graph to satisfy this property with high probability in as few rounds as possible.

We focus on the problem of constructing a Hamiltonian cycle in as few rounds as possible. In particular, we present an adaptive strategy for the player which achieves it in αn rounds, where $\alpha < 2.01678$ is derived from the solution to some system of differential equations. We also show that the player cannot achieve the desired property in less than βn rounds, where $\beta > 1.26575$. These results improve the previously best known bounds and, as a result, the gap between the upper and lower bounds is decreased from 1.39162 to 0.75102.

2012 ACM Subject Classification Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases Random graphs and processes, Online adaptive algorithms, Hamiltonian cycles, Differential equation method

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2022.29

Category RANDOM

Related Version *Full Version*: https://arxiv.org/abs/2205.02350 [10]

Acknowledgements The numerical results presented in this paper were obtained using the Julia language [5]. 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 Introduction and Main Results

1.1 Definitions

In this paper, we consider the **semi-random graph process** suggested by Peleg Michaeli, introduced formally in [3], and studied recently in [2, 9, 11, 1, 7, 13] that can be viewed as a "one player game". The process starts from G_0, the empty graph on the vertex set $[n] := \{1, \dots, n\}$ where $n \geq 1$. In each **step** t, a vertex u_t is chosen uniformly at random from $[n]$. Then, the player (who is aware of graph G_t and vertex u_t) must select a vertex v_t and add the edge $u_t v_t$ to G_t to form G_{t+1}. The goal of the player is to build a (multi)graph satisfying a given property $\mathcal{P}$ as quickly as possible. It is convenient to refer to u_t as a

[1] https://math.ryerson.ca/~pralat/research.html#publications

square, and v_t as a **circle** so every edge in G_t joins a square with a circle. We say that v_t is paired to u_t in step t. Moreover, we say that vertex $x \in [n]$ is **covered** by the square u_t arriving at round t, provided $u_t = x$. The analogous definition extends to the circle v_t. Equivalently, we may view G_t as a directed graph where each arc directs from u_t to v_t, and thus we may use (u_t, v_t) to denote the edge added in step t. For this paper, it is easier to consider squares and circles for counting arguments.

A **strategy** $\mathcal{S}$ is defined by specifying for each $n \geq 1$, a sequence of functions $(f_t)_{t=1}^{\infty}$, where for each $t \in \mathbb{N}$, $f_t(u_1, v_1, \ldots, u_{t-1}, v_{t-1}, u_t)$ is a distribution over $[n]$ which depends on the vertex u_t, and the history of the process up until step $t-1$. Then, v_t is chosen according to this distribution. If f_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 then denote $(G_i^{\mathcal{S}}(n))_{i=0}^{t}$ as the sequence of random (multi)graphs obtained by following the strategy $\mathcal{S}$ for t rounds; where we shorten $G_t^{\mathcal{S}}(n)$ to G_t or $G_t(n)$ when clear.

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

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

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

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

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

1.2 Main Results

In this paper, we concentrate on the property of having a Hamiltonian cycle, which we denote by $\texttt{HAM}$. As observed in [3], if G_t has a Hamiltonian cycle, then G_t has minimum degree at least 2. Thus, $\tau_{\texttt{HAM}} \geq \tau_{\mathcal{P}} = \ln 2 + \ln(1 + \ln 2) \geq 1.21973$, where $\mathcal{P}$ corresponds to having minimum degree 2. On the other hand, it is known that the famous 3-out process is Hamiltonian with probability tending to 1 as $n \to \infty$ (*a.a.s.*) [6]. As the semi-random process can be coupled with the 3-out process, we get that $\tau_{\texttt{HAM}} \leq 3$. A new upper bound was obtained in [9] in terms of an optimal solution to an optimization problem whose value is believed to be 2.61135 by numerical support. In the same paper, the lower bound mentioned above was shown to not be tight. The lower bound was increased by $\varepsilon = 10^{-8}$ and so numerically negligible.

The upper bound on $\tau_{\texttt{HAM}}$ of 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 above mentioned improvement proposed in [9] uses an adaptive strategy but in a weak sense. The strategy consists of 4 phases, each lasting a linear number of rounds, and the strategy is adjusted *only* at the end of each phase (for example, the player might identify vertices of low degree, and then focus on connecting circles to them during the next phase).

In this paper, we propose a fully adaptive strategy that pays attention to the graph G_t and the position of u_t for every single step t. As expected, such a strategy creates a Hamiltonian cycle substantially faster than our weakly adaptive strategy, and it allows us to improve the upper bound from 2.61135 to 2.01678.

▶ **Theorem 1.** *$\tau_{HAM} \leq \alpha \leq 2.01678$, where α is derived from a system of differential equations.*

Moreover, by investigating some specific structures that are generated by the semi-random process, which guarantee the existence of a large set of families of edges that cannot simultaneously contribute to the construction of a Hamiltonian cycle, we improve the lower bound of $\ln 2 + \ln(1 + \ln 2) \geq 1.21973$ to 1.26575. The structures we investigate in this work are different from the ones in [9]. We attain a simpler proof than in [9], and a much stronger bound.

▶ **Theorem 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, $\tau_{HAM} \geq \beta$.*

1.3 Previous Results

Let us briefly describe a few known results on the semi-random process. In the very first paper [3], it was shown that the process is general enough to approximate (using suitable strategies) several well-studied random graph models. 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 [1] where the conjecture from [3] was proven for any graph H: *a.a.s.* it takes at least $n^{(d-1)/d}/\omega$ rounds to create H.

Another property that was studied in the context of semi-random processes is a property of having a perfect matching, which we denote by PM. Since the 2-out process has a perfect matching *a.a.s.* [8], we immediately get that $\tau_{\texttt{PM}} \leq 2$. By coupling the semi-random process with another random graph that is known to have a perfect matching *a.a.s.* [12], the bound can be improved to $1 + 2/e < 1.73576$. This bound was recently improved by the authors of this paper by investigating another fully adaptive algorithm [11]. The currently best upper bound is $\tau_{\texttt{PM}} < 1.20524$. In the same paper, the lower bound observed in [3] ($\tau_{\texttt{PM}} \geq \ln(2) > 0.69314$) was improved as well, and now we know that $\tau_{\texttt{PM}} > 0.93261$ [11].

Finally, let us 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 [2], 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. They also proved that if $\Delta = \omega(\log(n))$, then this upper bound improves to $(\Delta/2 + o(\Delta))n$ rounds. Note that both of these upper bounds are asymptotic in Δ. When Δ is constant in n, such as in both the perfect matching and Hamiltonian cycle setting, determining the optimal dependence on Δ for the number of rounds needed to construct H remains open.

2 Proof of Theorem 1

2.1 Algorithmic Preliminaries

In this section, we introduce some notation as well as the basic ideas used in the design of all of our strategies.

The main ingredient for proving Theorem 1 is to specify a strategy which keeps augmenting or extending a path, until the path becomes Hamiltonian. Then, with a few more steps, the Hamiltonian path can be completed into a Hamiltonian cycle. Let us suppose that after $t \geq 0$ steps, we have constructed the graph G_t which contains the path P_t. Define U_t to be

the set of vertices *not* in P_t, which we refer to as the **unsaturated** vertices of $[n]$. It will be convenient to 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 $Q \subseteq V(P_t)$.

Let us first assume that u_{t+1} lands in U_t. In this case, we can clearly extend the path P_t by an edge by choosing v_{t+1} to be an endpoint of P_t. We call such a move a **(greedy) path extension**. Now, suppose that u_{t+1} lands on a vertex $x \in P_t$. In this case, we cannot perform a greedy path extension, however we can still choose v_{t+1} in a way that will help us extend the path in the future rounds. Specifically, set $v_{t+1} := r$ for some $r \in U_t$, and **colour** the vertex x as well as the edge xr. Suppose that in some round $i > t+1$, u_i lands on y next to the coloured vertex x on P_i (i.e., $d_{P_i}(x, y) = 1$). In this case, set $v_i = r$. Observe now that we can add r to the current path by adding the edges yr and xr to it, and by removing the edge yx. Thus, despite u_s not landing on an unsaturated vertex, we are still able to perform a move which extends its length by one. We call such an operation a **path augmentation**.

2.2 Proof Overview

In order to prove Theorem 1, we analyze a strategy which proceeds in three distinct **stages**. In the first stage, we execute `DegreeGreedy`, an algorithm which makes greedy path extensions whenever possible, and otherwise sets up path augmentation operations for future rounds in a degree greedy manner. Specifically, v_{t+1} is chosen amongst the unsaturated vertices of minimum coloured in-degree. 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 within the current stage). For the claimed (numerical) upper bound of Theorem 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, but the gain is rather insignificant. The second stage starts at some random step t_0 (i.e. $t_0 - 1$ is the total number of steps in stage one), and we execute `FullyRandomized`, an algorithm which makes greedy path extensions whenever possible, and otherwise chooses v_{t+1} randomly amongst the unsaturated vertices. We execute `FullyRandomized` until we are left with εn unsaturated vertices, where $\varepsilon = \varepsilon(n)$ tends to 0 as $n \to \infty$ arbitrarily slowly. At this point, we proceed to the final stage where a clean-up algorithm is run, which also uses path augmentations. Using elementary 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 $\alpha^* \leq 2.07721$. 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 $\tau_{\texttt{HAM}}$ of $\alpha^* \leq 2.07721$ (see Theorem 9). 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.

2.3 A Fully Randomized Algorithm

In order to describe our algorithm, it will be convenient to colour certain edges of G_t red. This helps us define certain vertices used by our algorithm for path augmentations. Specifically, $x \in V(P_t)$ 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. We denote the 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. 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) $|\mathcal{Q}_t| = |V(P_t)| - 5|\mathcal{L}_t|$.

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

Upon the arrival of u_{t+1}, there are four main cases our algorithm must handle. The first two cases involve extending the length of the path, whereas the latter two describe what to do when it is not possible to extend the path in the current round.

1. If u_{t+1} lands within U_t, then greedily extend P_t.
2. 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.
3. If u_{t+1} lands in $\mathcal{Q}_t$, then choose v_{t+1} u.a.r. amongst U_t, and colour $u_{t+1}v_{t+1}$ red. This case creates a one-red vertex.
4. 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_tv_t will not be used to construct a Hamiltonian cycle. In particular, the algorithm passes 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.

We now formally describe step $t+1$ of the algorithm when u_{t+1} is drawn u.a.r. from $[n]$. 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 $X(t) = |V(P_t)|$, $L_1(t) = |\mathcal{L}_t^1|$, $L_2(t) = |\mathcal{L}_t^2|$, and $L(t) = |\mathcal{L}_t| = L_1(t) + L_2(t)$. Note that $L(t)$ is an auxiliary random variable which we define only for convenience. We use Δ to denote the one step changes in our random variables (i.e., $\Delta X(t) := X(t+1) - X(t)$). Recall that t_0 is the step when `FullyRandomized` is called. Let us first show that our random variables cannot change drastically in one round.

▶ **Lemma 3** (Boundedness Hypothesis – `FullyRandomized`). *With probability* $1 - O(n^{-1})$,

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

for all $t_0 \leq t \leq 3n$ *with* $n - X(t) \geq n/\log n$.

Algorithm `FullyRandomized` Step $t+1$.

if $u_{t+1} \in U_t$ **then** ▷ greedily extend the path.
  Let v_{t+1} be an arbitrarily chosen endpoint of P_t.
  Set $V(P_{t+1}) = V(P_t) \cup \{u_{t+1}\}$, $E(P_{t+1}) = E(P_t) \cup \{u_{t+1}v_{t+1}\}$.
  Uncolour all of the edges adjacent to u_{t+1}.
else if $d_{P_t}(u_{t+1}, \mathcal{L}_t) = 1$ **then** ▷ path augment via red vertices
  Let $x \in \mathcal{L}_t$ be the (unique) red vertex adjacent to u_{t+1}
  Denote $xr \in E(G_t)$ an arbitrary red edge of x, and set $v_{t+1} = r$, where $r \in U_t$.
  Set $V(P_{t+1}) = V(P_t) \cup \{r\}$ and $E(P_{t+1}) = E(P_t) \cup \{u_{t+1}r, xr\} \setminus \{u_{t+1}x\}$.
  Uncolour all of the edges adjacent to r.
else if $u_{t+1} \in \mathcal{Q}_t \cup \mathcal{L}_t$ **then** ▷ construct red vertices
  Choose v_{t+1} u.a.r. from U_t.
  Colour $u_{t+1}v_{t+1}$ red. ▷ construct a one-red or two-red vertex
  Set $P_{t+1} = P_t$.
else ▷ pass on using edge $u_{t+1}v_{t+1}$.
  Choose v_{t+1} arbitrarily from $[n]$.
  Set $P_{t+1} = P_t$.
end if
Update $\mathcal{Q}_{t+1}$ such that $|\mathcal{Q}_{t+1}| = |V(P_{t+1})| - 5|\mathcal{L}_{t+1}|$.

Proof. Note that, by design, the path can only increase its length but it cannot absorb more than one vertex in each round. Hence, the desired property clearly holds for the random variable $X(t)$. 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 unsaturated vertex v. Observe that at any step $t \le 3n$, since we have assumed there are at least $n/\log n$ unsaturated vertices, the number of red edges adjacent to v is stochastically upper bounded by the binomial random variable $\mathrm{Bin}(3n, \log n/n)$ 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 $H_t = (X(i), L_1(i), L_2(i))_{0 \le i \le t}$. Note that H_t does *not* encompass the entire history of the random process after t rounds (i.e., $G_0, \dots, G_t$, the first $t+1$ graphs constructed by the algorithm). This deferred information exposure permits a tractable analysis of the random positioning of v_t when u_t is red. We observe the following expected difference equations.

▶ **Lemma 4** (Trend Hypothesis – `FullyRandomized`). *For each $t \ge t_0$, if $n - X(t) \ge n/\log n$, then*

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

$$\begin{aligned}\mathbb{E}[\Delta L_1(t) \mid H_t] = {} & \frac{X(t) - 5L(t)}{n} + \frac{2L_1(t)}{n}\left(\frac{2L_2(t)}{n - X(t)} - \frac{L_1(t)}{n - X(t)} - 1\right) \\ & + \frac{2L_2(t)}{n}\left(1 + \frac{2L_2(t)}{n - X(t)} - \frac{L_1(t)}{n - X(t)}\right) - \frac{L_1(t)}{n} \\ & + \left(1 - \frac{X(t)}{n}\right)\left(\frac{2L_2(t)}{n - X(t)} - \frac{L_1(t)}{n - X(t)}\right) + O(\log n/n)\end{aligned} \tag{2}$$

$$\mathbb{E}[\Delta L_2(t) \mid H_t] = \frac{L_1(t)}{n} - \left(1 - \frac{X(t)}{n}\right)\frac{2L_2(t)}{n - X(t)} - \frac{2L_1(t)}{n}\frac{2L_2(t)}{n - X(t)} - \frac{2L_2(t)}{n}\left(1 + \frac{2L_2(t)}{n - X(t)}\right) + O(\log n/n). \tag{3}$$

The proof is obtained by examining how the landing of u_t affects the random variables under study. For instance, for $X(t)$, observe that $\Delta X(t)$ is 1 when u_{t+1} lands on an unsaturated vertex, or adjacent to a red vertex; and is 0 otherwise. Combining with the probabilities of the above two events yields (1). The proofs for (2) and (3) are similar and the details can be found in Appendix A.

In order to analyze `FullyRandomized`, we shall employ the differential equation method [15]. 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 [4] for a gentle introduction to the methodology, and to Theorem 16 of Appendix C for a statement of the method which be sufficient for our purposes. The execution of `FullyRandomized` starts at some random step t_0, which we will prove is a.a.s. asymptotic to $s_0 n$ for some constant $0 \leq s_0 < 1$. Let $X(t_0)$ denote the number of vertices on P_t after the execution of `DegreeGreedy`. We shall prove that there exists some constant $\hat{x}(s_0)$ such that $|X(t_0)/n - \hat{x}(s_0)| \leq \lambda$ for some $\lambda = o(1)$. If N is set to 0, then $t_0 = s_0 = X(0) = \hat{x}(0) = 0$.

Let us now fix a sufficiently small constant $\varepsilon > 0$, and define the bounded domain

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

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

$$x' = 1 - x + 2(\ell_1 + \ell_2) \tag{4}$$

$$\ell_1' = x - 5(\ell_1 + \ell_2) + \ell_1\left(\frac{2\ell_2 - \ell_1}{1 - x} - 1\right) + 2\ell_2\left(1 + \frac{2\ell_2 - \ell_1}{1 - x}\right) - \ell_1 + 2\ell_2 - \ell_1 \tag{5}$$

$$\ell_2' = \ell_1 - 2\ell_2 - 2\ell_1\left(\frac{2\ell_2}{1 - x}\right) - 2\ell_2\left(1 + \frac{2\ell_2}{1 - x}\right). \tag{6}$$

The right-hand side (r.h.s.) of each of the above equations is Lipchitz on the domain $\mathcal{D}_\varepsilon$. Define

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

Now, the "Initial Condition" of Theorem 16 is satisfied with values $(s_0, \hat{x}(s_0), 0, 0)$ and some $\lambda = o(1)$. Moreover, the "Trend Hypothesis" and "Boundedness Hypothesis" are satisfied with some $\delta = O(\log n/n)$, $\beta = O(\log n)$ and $\gamma = o(n^{-1})$, by Lemmas 3 and 4. Thus, for every $\delta > 0$, $X(t) = nx(t/n) + o(n)$, $L_1(t) = n\ell_1(t/n) + o(n)$ 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, ℓ_1 and ℓ_2 are the unique solution to (4)–(6) with initial conditions $x(s_0) = \hat{x}(s_0)$, $\ell_1(s_0) = \ell_2(s_0) = 0$, and $\sigma(\varepsilon)$ is the supremum of s to which the solution can be extended before reaching the boundary of $\mathcal{D}_\varepsilon$. For $N = 0$, $s_0 = 0$ and the initial conditions are simply $x(0) = \ell_1(0) = \ell_2(0) = 0$. This immediately yields the following.

▶ **Lemma 5** (Concentration of `FullyRandomized`'s Random Variables). *For every $\delta > 0$, a.a.s. for all $t_0 \le t \le (\sigma(\varepsilon) - \delta)n$,*

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

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

$$\alpha^* := \lim_{\varepsilon \to 0+} \sigma(\varepsilon) \tag{7}$$

exists. It is obvious that $|L_1(t)/n|$ and $|L_2(t)/n|$ are both 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 6.** *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 $t \ge (\alpha^* - \delta)n$;*
- *$\alpha^* = 3$.*

The ordinary differential equations (4)–(6) do not have an analytical solution. In both cases $N = 0$ and $N = 100$, numerical solutions show that $\alpha^* < 2.1$. (For $N = 0$, $\alpha^* \approx 2.07721$.) Thus, by the end of the execution of `FullyRandomized`, there are εn unabsorbed vertices 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 assumption. 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 7** (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 8. 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. The proof is presented in Appendix A.

By setting $N = 0$ we immediately get an algorithm which a.a.s. constructs a Hamiltonian cycle in $\hat{\alpha}n$ steps, where $\hat{\alpha} \le 2.07721$. To obtain the better bound in Theorem 1, we set $N = 100$, and the execution of `DegreeGreedy` will be analysed in the next subsection.

▶ **Theorem 9.** *$\tau_{\texttt{HAM}} \le \hat{\alpha} \le 2.07721$, where $\hat{\alpha}$ is defined in (7) with initial conditions for (4)–(6) set by $x(0) = \ell_1(0) = \ell_2(0) = 0$.*

Proof. This follows by Proposition 6, the numerical value of α^*, and Lemma 7. ◀

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. As before, our algorithm uses path augmentations, and we colour the edges 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 red. 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. Once again, we denote our unsaturated vertices by U_t, and also maintain a set of **permissible** vertices $\mathcal{Q}_t$ which indicate which saturated vertices are allowed to be coloured blue. Specifically, using the same reasoning as before, we ensure the following:

(i) $|\mathcal{Q}_t| = |V(P_t)| - 5|\mathcal{L}_t|$.

(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 five main cases our algorithm must handle. The first two cases involve extending the length of the path, whereas the latter three describe what to do when it is not possible to extend the path in the current round.

1. If u_{t+1} lands within U_t, then greedily extend P_t.
2. 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.
3. 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.
4. 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.
5. 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} uniformly at random, 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. Note that red vertices are only created when the blue edges of magenta vertices are uncoloured as a side effect of path extensions and augmentations (see lines (4) and (14) below). We now formally describe step $t + 1$ of the algorithm upon receiving u_{t+1}:

Algorithm `DegreeGreedy` Step $t+1$.

if $u_{t+1} \in U_t$ **then** ▷ greedily extend the path
  Let v_{t+1} be an arbitrarily chosen endpoint of P_t.
  Set $V(P_{t+1}) = V(P_t) \cup \{u_{t+1}\}$, $E(P_{t+1}) = E(P_t) \cup \{u_{t+1}v_{t+1}\}$.
  Uncolour all of the edges adjacent to u_{t+1}.
else if $d(u_{t+1}, \mathcal{L}_t) = 1$ **then** ▷ path augment via coloured vertices
  Let $x \in \mathcal{L}_t$ be the (unique) coloured vertex adjacent to u_{t+1}
  if x is red **then**
    Denote $xy \in E(G_t)$ the red edge of x, where $y \in U_t$.
  else ▷ x is blue or magenta
    Denote $xy \in E(G_t)$ the blue edge of x, where $y \in U_t$.
  end if
  Set $v_{t+1} = y$.
  Set $V(P_{t+1}) = V(P_t) \cup \{y\}$ and $E(P_{t+1}) = E(P_t) \cup \{u_{t+1}y, xy\} \setminus \{u_{t+1}x\}$.
  Uncolour all of the edges adjacent to y.
else if $u_{t+1} \in \mathcal{Q}_t \cup \mathcal{R}_t$ **then** ▷ construct coloured vertices
  Choose v_{t+1} u.a.r. from the vertices of U_t of minimum blue degree.
  Colour $u_{t+1}v_{t+1}$ blue. ▷ create a blue or magenta vertex
  Set $P_{t+1} = P_t$.
else if $u_{t+1} \in \mathcal{B}_t$ **then**
  Choose v_{t+1} u.a.r. from U_t.
  Colour the edge $u_{t+1}v_{t+1}$ red. ▷ create a magenta vertex
  Set $P_{t+1} = P_t$.
else ▷ pass on using edge $u_{t+1}v_{t+1}$
  Choose v_{t+1} u.a.r. from $[n]$.
  Set $P_{t+1} = P_t$.
end if
Update $\mathcal{Q}_{t+1}$ such that $|\mathcal{Q}_{t+1}| = |V(P_{t+1})| - 5|\mathcal{L}_{t+1}|$. ▷ update permissible vertices

For each $t \geq 0$, define the random variables $X(t) := |V(P_t)|$, $B(t) := |\mathcal{B}_t|$, $R(t) := |\mathcal{R}_t|$, $M(t) := |\mathcal{M}_t|$ and $L(t) := |\mathcal{L}_t| = B(t) + R(t) + M(t)$. For each $q \geq 0$ and $t \geq 0$, 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 obvious 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 $\tau_{q-1} \leq t < \tau_q$. Observe that during phase q, each unsaturated vertex has blue degree $q-1$ or 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 $y \in U_t$ 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 an unsaturated 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 Section 3, we inductively define the functions x, r 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 10.** *A.a.s.* $\tau_q \sim \sigma_q n$ *for every* $0 \le q \le N$.[2] *Moreover, at step* τ_q, *a.a.s.*

$$X(\tau_q) \sim x(\sigma_q)n, \quad R(\tau_q) \sim r(\sigma_q)n,$$
$$C_{k_1,k_2}(\tau_q) \sim c_{k_1,k_2}(\sigma_q)n \quad \text{for all } (k_1,k_2) \text{ where } k_1+k_2 = q.$$

Although the method in the proof of Lemma 10 is similar to that of Lemmas 3, 4, 5 and Proposition 6, the analysis is much more intricate and involved. Before proving Lemma 10, we explain how we use it to prove Theorem 16.

Proof of Theorem 1. Set $N = 100$. By Lemma 10, the execution of `DegreeGreedy` ends at some step $t_0 \sim \sigma_N n$. Moreover, $X(t_0) \sim x(\sigma_N)n$. Numerical computation shows that $\sigma_N \approx 2.00189$. Next, the algorithm executes `FullyRandomized`. Let α^* be as defined in (7) where the initial conditions to the differential equations (4)–(6) are set by $s_0 = \sigma_N$, $x(s_0) = x(\sigma_N) \approx 0.99991$, and $\ell_1(s_0) = \ell_2(s_0) = 0$. Numerical computations show that $\alpha^* \approx 2.01678$. By Proposition 6 and the fact that $\alpha^* < 3$, the execution of the first two stages (`DegreeGreedy` and `FullyRandomized`) finishes at some step $(\alpha^* + o(1))n$, 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 7. The theorem follows. ◀

3 Proving Lemma 10

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

▶ **Lemma 11** (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* $1 - O(n^{-1})$,

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

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

Proof. Since $q \le N$ 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 3, and then union bound over all $k_1, k_2 \ge 0$ such that $k_1 + k_2 \in \{q-1, q\}$. ◀

We now state the conditional expected differences of our random variables. For space considerations, we defer their derivations to the full version of the paper [10].

Let H_t denote the history of the above random variables during the first t rounds. where we assume that $\tau_{q-1} \le t < \tau_q$ is such that $n - X(t) = \Omega(n)$. Firstly, observe that once again:

$$\mathbb{E}[\Delta X(t) \mid H_t] = 1 - \frac{X(t)}{n} + \frac{2L(t)}{n} + O(1/n) \tag{8}$$

We now consider $\Delta R(t)$:

$$\begin{aligned}\mathbb{E}[\Delta R(t) \mid H_t] = {} & \frac{M(t)}{n} - \frac{R(t)}{n} - \frac{2(B(t)+M(t))}{n}\frac{R(t)}{(n-X(t))} \\ & + \sum_{\substack{j,h: \\ j+h \in \{q-1,q\}}} \frac{2h(M_{j,h}(t)+B_{j,h}(t))}{n} \\ & - \frac{2R(t)}{n}\left(1 + \frac{R(t)}{n-X(t)} - \frac{M(t)}{n-X(t)}\right) - \frac{R(t)}{n} + O(1/n)\end{aligned} \tag{9}$$

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

Consider $\Delta C_{k_1,k_2}(t)$ and first assume that $k_1 + k_2 = q - 1$:

$$\begin{aligned}\mathbb{E}[\Delta C_{k_1,k_2}(t) \mid H_t] &= \frac{M_{k_1-1,k_2+1}(t)}{n} \cdot \mathbf{1}_{k_1>0} - \frac{C_{k_1,k_2}(t)}{n} - \frac{M_{k_1,k_2}(t)}{n} \\ &+ \frac{2(B(t)+M(t))}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)}\right) \\ &- \frac{2(B_{k_1,k_2}(t)+M_{k_1,k_2}(t))}{n} \\ &+ \frac{2R(t)}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)} - \frac{C_{k_1,k_2}(t)}{n-X(t)}\right) \\ &- \frac{(X(t)-5L(t))}{n}\frac{C_{k_1,k_2}(t)}{D(t)} \\ &+ \frac{B_{k_1+1,k_2-1}(t)}{n} \cdot \mathbf{1}_{k_2>0} - \frac{R(t)}{n}\frac{C_{k_1,k_2}(t)}{D(t)} - \frac{B_{k_1,k_2}(t)}{n} + O(1/n)\end{aligned} \tag{10}$$

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

$$\begin{aligned}\mathbb{E}[\Delta C_{k_1,k_2}(t) \mid H_t] &= \frac{M_{k_1-1,k_2+1}(t)}{n} \cdot \mathbf{1}_{k_1>0} - \frac{C_{k_1,k_2}(t)}{n} - \frac{M_{k_1,k_2}(t)}{n} \\ &+ \frac{2(B(t)+M(t))}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)}\right) \\ &- \frac{2(B_{k_1,k_2}(t)+M_{k_1,k_2}(t))}{n} \\ &+ \frac{2R(t)}{n}\left(\frac{M_{k_1-1,k_2+1}(t)}{n-X(t)} \cdot \mathbf{1}_{k_1>0} - \frac{M_{k_1,k_2}(t)}{n-X(t)} - \frac{C_{k_1,k_2}(t)}{n-X(t)}\right) \\ &+ \frac{(X(t)-5L(t))}{n}\frac{C_{k_1-1,k_2}(t)}{D(t)} \\ &+ \frac{B_{k_1+1,k_2-1}(t)}{n} \cdot \mathbf{1}_{k_2>0} + \frac{R(t)}{n}\frac{C_{k_1,k_2-1}(t)}{D(t)} - \frac{B_{k_1,k_2}(t)}{n} + O(1/n)\end{aligned} \tag{11}$$

We are now ready to prove Lemma 10. Firstly, when $q = 0$, by definition $\tau_0 = 0$, and so $\sigma_0 := 0$ trivially satisfies the conditions of Lemma 10. 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 and $c_{j,h}$ on $[0, \sigma_i]$ for each $j, h \geq 0$ with $j + h = i$, and Lemma 10 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 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 10 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

$$\mathcal{D}_\varepsilon := \left\{(s, x, r, (c_{j,h})_{j+h\in\{q-1,q\}}) : \sigma_{q-1} - 1 < s < 3, |x| < 1-\varepsilon, |r| < 2, |c_{j,h}| < 2, \varepsilon < \sum_{j,h:\, j+h=q-1} c_{j,h} < 2\right\}.$$

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 $d = \sum_{\substack{j,h:\\ j+h=q-1}} c_{j,h}$. Observe the following system of differential equations:

$$\begin{aligned} x' &= 1 - x + 2 \\ r' &= m - r - \frac{2(b+m)r}{1-x} + \sum_{\substack{j,h:\\ j+h\in\{q-1,q\}}} 2h(m_{j,h} + b_{j,h}) \end{aligned} \tag{12}$$

$$-2r\left(1+\frac{r}{1-x}-\frac{m}{1-x}\right)-r \tag{13}$$

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

$$\begin{aligned} c'_{k_1,k_2} &= m_{k_1-1,k_2+1}\cdot\mathbf{1}_{k_1>0}-c_{k_1,k_2}-m_{k_1,k_2} \\ &\quad +2(b+m)\left(\frac{m_{k_1-1,k_2+1}\cdot\mathbf{1}_{k_1>0}-m_{k_1,k_2}}{1-x}\right)-2(b_{k_1,k_2}+m_{k_1,k_2}) \\ &\quad +2r\left(\frac{m_{k_1-1,k_2+1}\cdot\mathbf{1}_{k_1>0}-m_{k_1,k_2}-c_{k_1,k_2}}{1-x}\right) \\ &\quad -(x-5\ell)\frac{c_{k_1,k_2}}{d}+b_{k_1+1,k_2-1}\cdot\mathbf{1}_{k_2>0}-r\frac{c_{k_1,k_2}}{d}-b_{k_1,k_2} \end{aligned} \tag{14}$$

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

$$\begin{aligned} c'_{k_1,k_2} &= m_{k_1-1,k_2+1}\cdot\mathbf{1}_{k_1>0}-c_{k_1,k_2}-m_{k_1,k_2} \\ &\quad +2(b+m)\left(\frac{m_{k_1-1,k_2+1}\cdot\mathbf{1}_{k_1>0}-m_{k_1,k_2}}{1-x}\right)-2(b_{k_1,k_2}+m_{k_1,k_2}) \\ &\quad +2r\left(\frac{m_{k_1-1,k_2+1}\cdot\mathbf{1}_{k_1>0}-m_{k_1,k_2}-c_{k_1,k_2}}{1-x}\right) \\ &\quad +(x-5\ell)\frac{c_{k_1-1,k_2}}{d}+b_{k_1+1,k_2-1}\cdot\mathbf{1}_{k_2>0}+r\frac{c_{k_1,k_2-1}}{d}-b_{k_1,k_2} \end{aligned} \tag{15}$$

The right-hand side (r.h.s.) of each of the above equations is Lipchitz on the domain $\mathcal{D}_\varepsilon$, as d is bounded below by ε. Define

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

By the inductive assumption, the "Initial Condition" of Theorem 16 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 (8)-(11). Finally, the "Boundedness Hypothesis" is satisfied with $\beta=O(\log n)$ and $\gamma=O(n^{-1})$ by Lemma 11. Thus, by Theorem 16, for every $\delta>0$, a.a.s. $X(t)=nx(t/n)+o(n)$, $R(t)=nr(t/n)+o(n)$ and $C_{k_1,k_2}(t)=nc_{k_1,k_2}(t/n)+o(n)$ uniformly for all $\sigma_{q-1}n\leq t\leq(\sigma(\varepsilon)-\delta)n$, where x, ℓ_1 and c_{k_1,k_2} are the unique solution to (12)-(15) 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 12** (Concentration of `DegreeGreedy`'s Random Variables). *For every $\delta>0$, a.a.s. for all $\tau_{q-1}\leq t\leq(\sigma(\varepsilon)-\delta)n$ and $k_1,k_2\geq 0$ such that $k_1+k_2\in\{q,q-1\}$,*

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

As $\mathcal{D}_\varepsilon\subseteq\mathcal{D}_{\varepsilon'}$ for every $\varepsilon>\varepsilon'$, $\sigma(\varepsilon)$ is monotonely nondecreasing as $\varepsilon\to 0$, and so $\sigma_q:=\lim_{\varepsilon\to 0+}\sigma(\varepsilon)$ exists. Moreover, the derivatives of the functions x,r, and c_{k_1,k_2} are uniformly bounded on (σ_{q-1},σ_q), which implies that the functions must be uniformly continuous. The latter condition implies that the functions are (uniquely) continuously extendable to $[\sigma_{q-1},\sigma_q]$, and so the following limits exist:

$$x(\sigma_q) := \lim_{s\to\sigma_q-}x(s) \tag{16}$$

$$r(\sigma_q) := \lim_{s\to\sigma_q-}r(s) \tag{17}$$

$$c_{k_1,k_2}(\sigma_q) := \lim_{s\to\sigma_q-}c_{k_1,k_2}(s). \tag{18}$$

Random variables $|R(t)/n|$ and $|C_{k_1,k_2}(t)/n|$ for $k_1+k_2 \in \{q, q-1\}$ are both bounded by 1 for all t. Thus, when t/n approaches σ_q, $X(t)/n$ approaches 1, or 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 13.** *For every $\varepsilon > 0$, there exists $\delta > 0$ such that a.a.s. one of the following holds.*

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

The ordinary differential equations (12)-(15) 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 (16)–(18) 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 10 is proven.

4 Proof of Theorem 2

Suppose G_t has a Hamiltonian cycle $H_t = H$ after $t \geq 0$ steps. Recall that for the (directed) semi-random edge (u_i, v_i), we refer to u_i as its square and v_i as its circle. We begin with the following observations:

1. H uses exactly n squares;
2. H uses at most 2 squares on each vertex;
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 obvious. For 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 2 above indicates the consideration of the random variable

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

which counts the total number of squares that can possibly contribute to H, truncated at 2 for each vertex. Observation 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 in 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 its first square in some step $i < t$, and y receives the corresponding circle in the same step;

(b) exactly one more square lands on x either before or after step i;
(c) at least two squares land on y after step s.

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.

Therefore, the total number of squares contributing to H is at most $Z - |\mathcal{W}_1| - |\mathcal{W}_2| + W$, where W accounts for double counting, which sometimes happens when there are $(x_1,y_1),(x_2,y_2) \in \mathcal{W}_1 \cup \mathcal{W}_2$ where $\{x_1,y_1\} \cap \{x_2,y_2\} \neq \emptyset$. More precisely, let

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

Then, $W := |\mathcal{T}_1| + |\mathcal{T}_2|$.[3]

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 [11, 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 of in-degree greater than μ 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 of in-degree at most μ 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 2–3 hold for approximate Hamiltonian cycles, and 1 holds for approximate Hamiltonian cycles with n replaced by $(1 - o(1))n$. Thus, no approximate Hamiltonian cycles can be built until step $Z - |\mathcal{W}_1| - |\mathcal{W}_2| + W \geq (1 - o(1))n$. 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 $\mathcal{S}$.

[3] Note that the cases where $((x_1,y_1),(x_2,y_2)) \in \mathcal{W}_1 \times \mathcal{W}_1$ such that $y_1 = y_2$ and $((x_1,y_1),(x_2,y_2)) \in \mathcal{W}_2 \times \mathcal{W}_2$ such that $y_1 = x_2$ do not cause double counting.

▶ **Lemma 14.** *Suppose $\mathcal{S}$ is μ-well-behaved. For every $t = \Theta(n)$, the following a.a.s. 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 in Theorem 2.

Proof of Theorem 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, $\tau_{\texttt{HAM}} \geq \beta$. ◀

5 Conclusion and Open Problems

We have made significant progress on reducing the gap between the previous best upper and lower bounds on $\tau_{\texttt{HAM}}$. 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. Rather, in the terminology of `DegreeGreedy`, when a (second) square lands on a blue vertex, the edge is greedily chosen amongst unsaturated vertices of minimum blue degree (opposed to u.a.r.). Unfortunately, it seems challenging to analyze this algorithm via the differential equation method, but it is likely that this algorithm will lead to an improved upper bound on $\tau_{\texttt{HAM}}$ of less than 2.

Another direction is to understand which graph properties exhibit **sharp thresholds**. It is known that for basic properties, such as minimum degree $k \geq 1$, sharp thresholds do exist [3]. Moreover, in [2] it was shown that if H is a spanning graph with max 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., $\Delta = O(\log n)$). Very recently, Surya and the second author [13], a developed a general machinery for proving the existence of sharp thresholds in adaptive random graph processes. Applied to the semi-random graph process, they show that sharp thresholds exist for the property of being Hamiltonian as well as to containing a perfect matching. This provides some evidence that sharp thresholds *do* exist when $\Delta = O(\log n)$, and we leave this an interesting open problem.

References

1 Natalie Behague, Trent Marbach, Paweł Prałat, and Andrzej Ruciński. Subgraph games in the semi-random graph process and its generalization to hypergraphs. *arXiv preprint*, 2022. doi:10.48550/ARXIV.2105.07034.

2 Omri Ben-Eliezer, Lior Gishboliner, Dan Hefetz, and Michael Krivelevich. Very fast construction of bounded-degree spanning graphs via the semi-random graph process. *Proceedings of the 31st Symposium on Discrete Algorithms (SODA'20)*, pages 728–737, 2020.

3 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.

4 Patrick Bennett and Andrzej Dudek. A gentle introduction to the differential equation method and dynamic concentration. *CoRR*, 2020. doi:10.48550/ARXIV.2007.01994.

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

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

7 Sofiya Burova and Lyuben Lichev. The semi-random tree process, 2022. doi:10.48550/ARXIV.2204.07376.

8 Alan M Frieze. Maximum matchings in a class of random graphs. *Journal of Combinatorial Theory, Series B*, 40(2):196–212, 1986.

9 Pu Gao, Bogumił Kamiński, Calum MacRury, and Paweł Prałat. Hamilton cycles in the semi-random graph process. *European Journal of Combinatorics*, 99:103423, 2022. doi:10.1016/j.ejc.2021.103423.

10 Pu Gao, Calum MacRury, and Pawel Pralat. A fully adaptive strategy for hamiltonian cycles in the semi-random graph process. *CoRR*, abs/2205.02350, 2022. doi:10.48550/ARXIV.2205.02350.

11 Pu Gao, Calum MacRury, and Paweł Prałat. Perfect matchings in the semi-random graph process. *SIAM Journal on Discrete Mathematics*, in press, 2022.

12 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, March 2020. doi:10.1016/j.jctb.2020.03.004.

13 Calum MacRury and Erlang Surya. Sharp thresholds in adaptive random graph processes. *CoRR*, 2022.

14 Lutz Warnke. On wormald's differential equation method. *CoRR*, abs/1905.08928, 2019. doi:10.48550/ARXIV.1905.08928.

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

A Proofs of Lemmas 4 and 7

Proof of Lemma 4. As discussed, `FullyRandomized` 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 that all our explanations below assume that we have conditioned on H_t. We focus on the second and third expected differences, where we make use of the following crucial observation:

1. Conditional on H_t, the circles of the red edges of $\mathcal{L}_t$ are distributed u.a.r. amongst the unsaturated vertices U_t.

Note that were we to condition on the full history, i.e., $G_0, \dots, G_t$, then these circles would be determined by the history of the process, and so the only randomness in the expectations would be over the draw of u_{t+1}. By averaging over this additional randomness, we are able to get the claimed expected differences.

Consider now the second expected difference and assume that u_{t+1} lands on an unsaturated vertex. Firstly, observe that this event occurs with probability $1 - X(t)/n$. On the other hand, all the red edges adjacent to u_{t+1} will be uncoloured after the path augmentation involving u_{t+1} is made. Now, because of 1., there are $\frac{L_1(t)}{n-X(t)}$ red edges belonging to $\mathcal{L}_t^1$ which are adjacent to u_{t+1} in expectation. After the path augmentation involving u_{t+1}, these edges are uncoloured and so $\frac{L_1(t)}{n-X(t)}$ one-red vertices are destroyed in expectation. Now, in expectation there are also $\frac{2L_2(t)}{n-X(t)} + O(\log n/n)$ red edges adjacent to u_{t+1} which belong to *distinct* two-red vertices. To see this, fix a two-red vertex $x \in \mathcal{L}_t^2$ and observe that because of 1., precisely one red edge of x is adjacent to u_{t+1} with probability $\frac{2}{n-X(t)} - \frac{1}{(n-X(t))^2}$. Since $n - X(t) \geq n/\log n$ by assumption, this probability is $\frac{2}{n-X(t)} + O((\log n/n)^2)$, and so the $\frac{2L_2(t)}{n-X(t)} + O(\log n/n)$ term follows after summing over all the vertices of $\mathcal{L}_t^2$. Now, after the

path augmentation involving u_{t+1}, these red edges are uncoloured. Since these red edges belonged to distinct two-red vertices, the path augmentation creates $\frac{2L_2(t)}{n-X(t)} + O(\log n/n)$ new one-red vertices in expectation. These two cases explain the $\left(1 - \frac{X(t)}{n}\right)\left(\frac{2L_2(t)}{n-X(t)} - \frac{L_1(t)}{n-X(t)}\right)$ term.

Let us now consider when u_{t+1} lands on a saturated vertex and $d_{P_t}(u_{t+1}, \mathcal{L}_t) = 1$, where x is the unique red vertex adjacent to u_{t+1}. If x is a one-red vertex, then let r be such that xr is the red edge of x. Observe that after the augmentation, xr will be uncoloured, and x will no longer be a red vertex. Moreover, in expectation there are $\frac{L_1(t)}{n-X(t)} + O(\log n/n)$ other red edges belonging to $\mathcal{L}_t^1$ which will be uncoloured. Thus, $1 + \frac{L_1(t)}{n-X(t)} + O(\log n/n)$ one-red vertices will be destroyed in expectation. On the other hand, there are $\frac{2L_2(t)}{n-X(t)} + O(\log n/n)$ red edges adjacent to r which belong to distinct two-red vertices in expectation. Thus, $\frac{2L_2(t)}{n-X(t)} + O(\log n/n)$ two-red vertices will become one-red vertices in expectation after augmenting via xr and $u_{t+1}r$. Since u_{t+1} lands next to a one-red vertex with probability, $\frac{2L_1(t)}{n}$, this explains the $\frac{2L_1(t)}{n}\left(\frac{2L_2(t)}{n-X(t)} - \frac{L_1(t)}{n-X(t)} - 1\right)$ term. An analogous argument explains the $\frac{2L_2(t)}{n}\left(1 + \frac{2L_2(t)}{n-X(t)} - \frac{L_1(t)}{n-X(t)}\right)$ term.

Consider when u_{t+1} lands in $\mathcal{Q}_t$. Observe that this occurs with probability $\frac{|\mathcal{Q}_t|}{n} = \frac{X(t)-5L(t)}{n}$. In this case, v_{t+1} is chosen u.a.r. amongst U_t and $u_{t+1}v_{t+1}$ is coloured red. Thus, u_{t+1} becomes a red vertex, and so $L_1(t)$ increases by 1 and we get $\Delta L_1(t) = 1$. This explains the $\frac{X(t)-5L(t)}{n}$ term.

The final case to consider is when u_{t+1} lands on a saturated vertex, and $u_{t+1} \in \mathcal{L}_t^1$. Observe that this occurs with probability $\frac{L_1(t)}{n}$. Moreover, the algorithm will then choose v_{t+1} u.a.r. amongst U_t and colour the edge $u_{t+1}v_{t+1}$ red. After this move, u_{t+1} will be converted from a one-red vertex to a two-red vertex, and so $\Delta L_1(t) = 1$. This explains the $\frac{-L_1(t)}{n}$ term.

By combining the contributions from all of the above cases, we get the second expected difference. The third expected difference follows via an analogous argument. ◀

Proof of Lemma 7. 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 $j_k = 0$. Obviously, $\tau_1 = O(\log n)$ and $\tau = O(n^{1/4})$.

We use a cleaning-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 in the end. The k-th iteration of the cleaning-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 $k \leq \tau_1$ and $m_k := n^{1/2}$ if $\tau_1 < k \leq \tau$. 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. This edge u_tv_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_tv_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 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 where 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, $m_k \geq n^{1/2}$. Indeed, this is obviously true for $\tau_1 < k \leq \tau$. On the other hand, if $k \leq \tau_1$, then $j_k = \varepsilon n(1/2)^k$ and so $m_k = \sqrt{nj_k} \geq \sqrt{n}$ (in fact, $m_k = \Omega(n^{5/8})$).

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

First, we consider iterations $k \leq \tau_1$. 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 each unsaturated vertex is adjacent to at most $R_k/j_{k-1} + 1 \leq m_k/j_{k-1} + 1$ red vertices. Moreover, exactly $j_{k-1} - j_k = (1/2)j_{k-1}$ vertices are absorbed in step (iii). As a result,

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

as $j_{k-1} = 2j_k \leq 2\sqrt{\varepsilon}m_k \leq 0.1m_k$. 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, 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

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

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

Next, consider iterations $\tau_1 < k \leq \tau$. 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 that with probability at least $1 - n^{-1}$, $R_k \geq 0.9m_k$. 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 $1 - n^{-1}$, $T_k \leq n^{1/2} + n^{1/2}\log^2 n \leq 2n^{1/2}\log^2 n$.

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

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

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. First, add $n^{1/2}$ semi-random edges u_tv_t where v_t is always u, discarding any multiple edges that could possibly be created. For each such semi-random edge u_tu, colour the left neighbour of u_t on P blue. Next, add add $n^{1/2}\log^2 n$ semi-random edges u_tv_t where v_t is always v. Suppose that some $u_t = x$ is blue. 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 Chernoff bound, a.a.s. a semi-random edge added during the second round hits a blue vertex, completing the proof. ◀

B Proof of Lemma 14

▶ **Lemma 15.** *Suppose $\mathcal{S}$ is μ-well-behaved up until time $t = \Theta(n)$. A.a.s. the following holds for $G_t^{\mathcal{S}}$:*

$$|\mathcal{W}_1| \sim n \sum_{i \le t} \frac{1}{n}\left(1-\frac{1}{n}\right)^t \sum_{i \le j_1 < j_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^{j_2} \tag{19}$$

$$|\mathcal{W}_2| \sim n \sum_{i_1 \le i_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^t \left(\sum_{i_1 \le j_1 < j_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^{j_2} + \sum_{i_2 < j_1 < j_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^{j_2} \right) \tag{20}$$

$$\begin{aligned} |\mathcal{T}_1| \sim{}& n \sum_{i \le t} \frac{1}{n}\left(1-\frac{1}{n}\right)^t \sum_{i \le j_1 < j_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^t \\ &\times \left(\sum_{j_1 \le h_1 < h_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^{h_2} + \sum_{j_2 < h_1 < h_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^{h_2} \right) \end{aligned} \tag{21}$$

$$\begin{aligned} |\mathcal{T}_2| \sim{}& n \sum_{i_1 < i_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^t \sum_{i_1 \le j_1 < j_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^t \\ &\times \left(\sum_{j_1 \le h_1 < h_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^{h_2} + \sum_{j_2 < h_1 < h_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^{h_2} \right) \\ &+ \sum_{i_1 < i_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^t \sum_{i_2 \le j_1 < j_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^t \\ &\times \left(\sum_{j_1 \le h_1 < h_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^{h_2} + \sum_{j_2 < h_1 < h_2 \le t} \frac{1}{n^2}\left(1-\frac{1}{n}\right)^{h_2} \right). \end{aligned} \tag{22}$$

Proof. We prove (19) and briefly explain the expressions in (20)–(22) whose proofs are similar to that of (19). 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 with 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 $n^{-2}(1-1/n)^{j_2-2} \sim n^{-2}(1-1/n)^{j_2}$. 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

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

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 (20)–(22):

In (20), 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 (21), 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 (22), 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. ◀

From Lemma 15, we deduce that for $t = sn$,

$$
\begin{aligned}
|\mathcal{W}_1| &\sim ne^{-s}\int_0^s dx\int_x^s dy_1\int_{y_1}^s e^{-y_2}dy_2 = ne^{-s}\left(1-\frac{e^{-s}s^2}{2}-e^{-s}s-e^{-s}\right)\\
|\mathcal{W}_2| &\sim ne^{-s}\int_0^s dx_1\int_{x_1}^s dx_2\left(\int_{x_1}^s dy_1\int_{y_1}^s e^{-y_2}dy_2+\int_{x_2}^s dy_1\int_{y_1}^s e^{-y_2}dy_2\right)\\
&= ne^{-s}\left(s-e^{-s}s^2-\frac{e^{-s}s^3}{2}-e^{-s}s\right)\\
|\mathcal{T}_1| &\sim ne^{-2s}\int_0^s dx\int_x^s dy_1\int_{y_1}^s dy_2\left(\int_{y_1}^s dz_1\int_{z_1}^s e^{-z_2}dz_2+\int_{y_2}^s dz_1\int_{z_1}^s e^{-z_2}dz_2\right)\\
&= ne^{-2s}\left(-1+s-\frac{e^{-s}s^3}{3}-\frac{e^{-s}s^2}{2}-\frac{e^{-s}s^4}{8}+e^s\right)\\
|\mathcal{T}_2| &\sim ne^{-2s}\int_0^s dx_1\int_{x_1}^s dx_2\int_{x_1}^s dy_1\int_{y_1}^s dy_2\left(\int_{y_1}^s dz_1\int_{z_1}^s e^{-z_2}dz_2+\int_{y_2}^s dz_1\int_{z_1}^t e^{-z_2}dz_2\right)\\
&\quad+ne^{-2s}\int_0^s dx_1\int_{x_1}^s dx_2\int_{x_2}^s dy_1\int_{y_1}^s dy_2\left(\int_{y_1}^s dz_1\int_{z_1}^s e^{-z_2}dz_2+\int_{y_2}^s dz_1\int_{z_1}^s e^{-z_2}dz_2\right)\\
&= ne^{-2s}\left(-s+s^2-e^{-s}s\left(\frac{s^4}{8}+\frac{s^3}{3}+\frac{s^2}{2}-1\right)\right)
\end{aligned}
$$

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

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

C The Differential Equation Method

In this section, we provide a self-contained *non-asymptotic* statement of the differential equation method. The statement combines [14, Theorem 2], and its extension [14, Lemma 9], in a form convenient for our purposes, where we modify the notation of [14] slightly. In particular, we rewrite [14, 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 3 and 11.

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 16** (Differential Equation Method, [14]). *Suppose we are given σ-fields $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots$, and for each $t \geq 0$, random variables $((Y_k(t))_{1\leq k\leq a}$ which are $\mathcal{F}_t$-measurable. Define $T_{\mathcal{D}}$ to be the minimum $t \geq 0$ such that*

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

Let $T \geq 0$ be an (arbitrary) stopping time[4] adapted to $(\mathcal{F}_t)_{t\geq 0}$, and assume that the following conditions hold for $\delta, \beta, \gamma \geq 0$ and $\lambda \geq \delta \min\{S, L^{-1}\} + R/n$:

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

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

(ii) *The "Trend Hypothesis": For each $t \leq \min\{T, T_{\mathcal{D}} - 1\}$,*

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

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

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

for each $t \leq \min\{T, T_{\mathcal{D}} - 1\}$:

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

$$\max_{0\leq t\leq \min\{T,\sigma n\}} \max_{1\leq k\leq a} |Y_k(t) - y_k(t/n)n| < 3\lambda \exp(LS)n, \tag{23}$$

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

$$y_k'(s) = F_k(s, y_1(s), \ldots, y_a(s)) \quad \text{with } y_k(0) = \hat{y}_k \text{ for } 1 \leq k \leq a, \tag{24}$$

and $\sigma = \sigma(\hat{y}_1, \ldots, \hat{y}_a) \in [0, S]$ is any choice of $\sigma \geq 0$ 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 $s \in [0, \sigma)$.

▶ Remark 17. Standard results for differential equations guarantee that (24) has a unique solution $(y_k(s))_{1\leq k\leq a}$ which extends arbitrarily close to the boundary of $\mathcal{D}$.

▶ Remark 18. The proof of Theorem 16 works for any choice of $R \in [1, \infty)$ and $T \in (0, \infty)$ which satisfy $\max_{1\leq k\leq a} |F_k(x)| \leq R$ for all $x = (s, y_1, \ldots, y_a) \in \mathcal{D}$ and $0 \leq s \leq T$.

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