Contents

Abstract

Let G be a random graph. It is known that if G has at least $\frac{n}{2}(\log n + \log\log n)$ edges, and if it has minimum degree 2, then G is almost surely Hamiltonian.

Several polynomial time algorithms were suggested in order to find the Hamiltonian cycles in such graphs, among them there is the algorithm HAM, which was suggested by Bollobás Fenner and Frieze.

In this work we ask what if the input for HAM was a semi random graph instead of a random graph. A semi random graph is constructed by selecting a random graph and then adding to it arbitrary edges.

We discuss several problems that may occur in this case. Then we suggest a polynomial time algorithm Semi-Ham, which is an adaptation of HAM to the semi random case.

Chapter 1

Introduction

1.1 The Problem of Hamiltonicity

The problem of Hamiltonicity is defined as follows: given a graph $G(V, E)$, either directed or undirected, does G contain a Hamiltonian cycle, that is, a simple cycle that passes through all its vertices. If so, we say that G is Hamiltonian. The problem of Hamiltonicity is known to be NP-complete.

One way of coping with NP-hard problems is to consider them from a probabilistic point of view. Informally, using this approach, we develop and analyze algorithms that would work well for the average case instances. This enables us to avoid the worse case instances, which are usually used for NP-reductions, and hence, for some problems, we can hope to achieve good results for most of the instances.

Consider the problem of Hamiltonicity. Assume we are given an algorithm for finding Hamiltonian cycles and we want to analyze its performance on some class of graphs. To do this analysis we can select a random graph from this class. If the probability that the algorithm succeeds on the random graph tends to 1, then it will succeed on most of the graphs in the class. On the other hand, if the probability tends to 0, then it will fail on most of the graphs in the class.

A related question is as follows. Given a class of graphs, are most of the graphs in the class are Hamiltonian or not. That is, we are only interested in the existence question.

For example, let F_m be the class of graphs with n vertices and m edges. We then can ask what are the families for which most of the graphs are Hamiltonian. To answer this type of questions, we take a random graph with m edges, and ask what is the probability that it is Hamiltonian. Then if the probability tends to 1, then most of the graphs with m edges are Hamiltonian, and if the probability tends to 0 then most of the graphs with m edges are not Hamiltonian. This question is discussed in Section 2.2.2

1.2 Structure of the Work

The two basic notions of random graphs and semi-random graphs are discussed in Chapter 2 and in Chapter 3. In Chapter 4 we discuss the algorithm HAM of Bollobás, Fenner and Frieze, which finds Hamiltonian cycles in random graphs, and discuss the problems it might have on semi-random input instances. To solve these problem, we suggest the algorithm Semi-Ham. We first assume that the random part of the inputs have minimum degree 3. Under this assumption, we describe the algorithm Semi-Ham in Chapter 5, and analyze its performance in Chapter 6. In Chapter 7 we extend the algorithm and its analysis to the general case. Summary is given in Chapter 8.

1.3 Some Basic Notation

We give here some basic notation which are used throughout the work.

Graph Theory

Let $G(V,E)$ be a graph. Then $n = |V|$, and $m = |E|$, and $N = \binom{n}{2}$ is the number of edges in the complete graph with n vertices. Two graph $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are *isomorphic* if there exists a bijection $\phi : V_1 \to V_2$ such that $(x,y) \in E_1$ iff $(\phi(x), \phi(y)) \in E_2$. A graph $G'(V', E')$ is a *spanning subgraph* of a graph $G(V,E)$ if $V = V'$, and $E' \subseteq E$.

For a set of vertices $S \subseteq V$, $E(S)$ is the set of edges spanned by S, and $e(S) = |E(S)|$. Similarly, for two sets $S_1, S_2 \subseteq V$, $E(S_1, S_2)$ is the set of edges with one endpoint in S_1, and the other endpoint in S_2, $e(S_1, S_2) = |E(S_1, S_2)|$.

For a path P, $V(P)$ is the set of vertices in P, and $E(P)$ is the set of edges. We use the notation '$\circ$', for concatenation of paths or edges. That is, $P \circ e$, is the concatenation of the path P and the edge e, in this case they must share a common endpoint. Whenever we index a path with two indices, then these indices are its endpoints, for example P_{v_1,v_r} is a path whose endpoints are v_1 and v_r.

For a set of vertices S, $N(S)$ is the neighborhood of S, that is, $N(S) = \{v \notin S, \exists (v,w) \in E, w \in S\}$, and $N^*(S) = N(S) \cup S$. For a vertex v, $N(v) = N(\{v\})$ and $N^*(v) = N^*(\{v\})$. The degree of a vertex $\deg(v) = |N(v)|$. The minimum degree in the graph is $\delta(G)$.

Asymptotic Notation

Let $f(n), g(n)$ be two non negative functions.

$f(n) = O(g(n))$ if there exist constants $C > 0$ and n_0, such that $f(n) \leq C \cdot g(n)$ for all $n \geq n_0$.

$f(n) = \Omega(g(n))$ if there exist constants $C > 0$ and n_0, such that $f(n) \geq C \cdot g(n)$ for all $n \geq n_0$.

$f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

$f(n) = o(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.

$f(n) = \omega(g(n))$ if $g(n) = o(f(n))$.

$f(n) \ll g(n)$ if $f(n) = o(g(n))$.

Probability

A sequence of events $\mathcal{E}_n$ occurs with high probability (whp), if

$$\lim_{n\to\infty} \mathbb{P}[\mathcal{E}_n] = 1 \ ,$$

in this case, we also say that $\mathcal{E}_n$ almost surely (a.s.) occurs.

For a probability space Ω and an event E, we say that almost every (a.e.) $\omega \in \Omega$ is in E, if $\lim_{n\to\infty} \mathbb{P}[\omega \in E] = 1$.

Chapter 2

Random Graphs

2.1 Models of Random Graphs

Every probability space, whose points are graphs, would define us the notion of random graph. We would use two basic models in order to define these probability spaces. The first one is the $\mathcal{G}_{n,m}$ model, and the second one is the $\mathcal{G}_{n,p}$ model.

In the first model, a probabilistic space $\mathcal{G}_{n,m}$ is defined for $0 \leq m \leq N = \binom{n}{2}$. The space $\mathcal{G}_{n,m}$ consists all $\binom{N}{m}$ graphs with n vertices and m edges. All elements (graphs) in the space are taken to be equiprobable. We write $G_{n,m}$ for a random graph in the space $\mathcal{G}_{n,m}$. For a fixed graph G with m edges, we have,

$$\mathbb{P}[G_{n,m} = G] = \binom{N}{m}^{-1}$$

In the second model, a probabilistic space $\mathcal{G}_{n,p}$ is defined for $0 \leq p \leq 1$. A random graph $G_{n,p}$ in the space $\mathcal{G}_{n,p}$ is defined as a graph with n vertices, where each of the possible N edges is independently included with probability p. For a fixed graph $G(V, E)$, we have,

$$\mathbb{P}[G_{n,p} = G] = p^{|E|}(1-p)^{N-|E|}$$

Another model is the model of random graph processes, this is discussed in Section 2.2.3.

2.1.1 Graph Properties and Threshold Functions

A graph *property* is a set of graphs closed under isomorphism. A property Q is called *increasing* if it is preserved when adding edges. A property Q is called *decreasing* if it is preserved when deleting edges. The complement of an increasing property is a decreasing property and vise versa. A property that is either increasing or decreasing is called *monotone*. For example, the property of Hamiltonicity is increasing.

Let Q be an increasing graph property. For the $\mathcal{G}_{n,m}$ model, a function $\hat{m} = \hat{m}(n)$ is a *threshold* function if

$$\lim_{n\to\infty} \mathbb{P}[G_{n,m} \in Q] = \begin{cases} 0 & \text{if } m \ll \hat{m} \\ 1 & \text{if } m \gg \hat{m} \end{cases}$$

Analogously, for the $\mathcal{G}_{n,p}$ model, a function $\hat{p} = \hat{p}(n)$ is a *threshold* function if

$$\lim_{n\to\infty} \mathbb{P}[G_{n,p} \in Q] = \begin{cases} 0 & \text{if } p \ll \hat{p} \\ 1 & \text{if } p \gg \hat{p} \end{cases}$$

In a similar way, we can define a threshold function for decreasing properties. Every monotone property has a threshold function (see [15, Theorem 1.24]).

A threshold function is not unique. If $\hat{m}$ (or $\hat{p}$) is a threshold function, then for any constant $c > 0$, $c\hat{m}$ ($c\hat{p}$), is also a threshold function. For some monotone properties we can show the existence of a *sharp threshold*, that is, a function $\hat{p}$ such that,

$$\lim_{n\to\infty} \mathbb{P}[G_{n,p} \in Q] = \begin{cases} 0 & \text{if } p < (1-\alpha)\hat{p} \\ 1 & \text{if } p > (1+\alpha)\hat{p} \end{cases}$$

for every constant $\alpha > 0$. A sharp threshold is defined analogously in the $\mathcal{G}_{n,m}$ model. For our case, the property of Hamiltonicity has a sharp threshold, this is discussed in Section 2.2.2.

When proving results concerning random graphs, it is usually convenient to work with the $\mathcal{G}_{n,p}$ model, because of the independency of choosing the edges. The following claim states that if $m = Np$ then we can apply these results for the $\mathcal{G}_{n,m}$ model.

Claim 2.1.1 *Let Q be a monotone property, and let $p(n) = m(n)/N$, $m(n) \to \infty$. Then,*

(1) if $\mathbb{P}[G_{n,p} \in Q] \to 1$ then $\mathbb{P}[G_{n,m} \in Q] \to 1$

(2) if $\mathbb{P}[G_{n,p} \in Q] \to 0$ then $\mathbb{P}[G_{n,m} \in Q] \to 0$

(3) if $p(n)$ is a threshold function for Q in the $\mathcal{G}_{n,p}$ model, then $m(n)$ is a threshold function for Q in the $\mathcal{G}_{n,m}$ model.

Similar assertions hold in the opposite direction, but they are not required for our work.

2.2 Hamiltonicity and Random Graphs

2.2.1 The Extension-Rotation Technique

We start this section with a review of the extension-rotation technique, which is a basic concept in many papers about Hamiltonicity and random graphs (including this work).

The key idea is as follows. We have a path P, and we try to extend it. If we could not extend it, then we use P to generate another path P' of the same length and with the same set of vertices. We then try to extend P' and so on.

We start with a description of the three basic operations of the technique. Let $P = (v_1, \dots, v_r)$ be a path of length $r < n$. The first two operations, simple extension and cycle extension are used in order to generate a path of length $r + 1$.

Simple extension : If v_r has a neighbor $w \notin P$, then we can extend the path P by adding the edge (v_r, w).

Cycle extension : Assume $2 < r$. If there is an edge (v_1, v_r) in the graph, then by adding this edge we turn P into a cycle C of length r. If the graph is connected, then there must be a vertex v_j in C, which has a neighbor $w \notin C$. Then, by deleting (v_j, v_{j+1}) and adding (v_j, w), we get a path $P' = (w, v_j, v_{j-1}, \dots, v_1, v_r, v_{r-1}, \dots, v_{j+1})$ of length $r + 1$.

For the last operation, rotation, we assume that for the path P we could not perform simple or cycle extension. We then use rotation, in order to generate another path starting with v_1 of the same length.

Rotation : Since no cycle extension neither simple extension were done, then $N(v_r) \subseteq \{v_2, \dots, v_{r-1}\}$. Let v_j, be a neighbor of v_r which is not v_{r-1} (if there is one). Then by deleting the edge (v_j, v_{j+1}) from P, and adding the edge (v_j, v_r) to P, we get a new path $P' = (v_1, \dots, v_j, v_r, v_{r-1}, \dots, v_{j+1})$ of length r. In this case, we say that v_1 is the *fixed endpoint*, v_j is the *pivot*, (v_j, v_{j+1}) is the *broken edge*, and (v_j, v_r) is the *rotation edge*. We denote this operation by $P' = \text{Rotate}(P, (v_j, v_r))$, and we say that P' is generated from P by a single rotation.

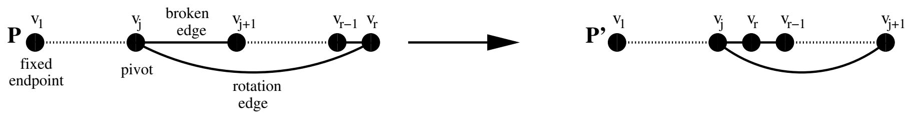

Figure 2.1: $P' = \text{Rotate}(P, (v_j, v_r))$

Observe, that any path P'' that is generated from P' by a single rotation, can be generated from P by at most two rotations. In general, if some path Q is generated from a path P by t rotations, then any path that is generated from Q by a single rotation, can be generated from P by at most $t + 1$ rotations.

The exact details of the implementation of the extension-rotation technique may vary from paper to paper. We give here some examples.

- Given a path P, we can try to extend or rotate it from both its endpoints (as it is done in HAM, see Section 4.1), or work with one fixed endpoint (as it is done in our algorithm Semi-Ham).

- We can bound the number of rotations from the initial path. This is usually done when the technique is used for an algorithm, in order to keep the running time polynomial.
- We can disallow some of the possible rotations. For example, we can force the rotation edges to be of a given subset of edges (as it is done in [6]).

2.2.2 Threshold for Hamiltonicity in Random Graphs

The threshold for Hamiltonicity was an open problem, presented by Erdôs and Rényi in [8]. The first breakthrough was done by Pósa in [19]. Pósa showed that for a random graph $G_{n,m}$, if $m(n) > cn \log n$, for some large constant c, then $G_{n,m}$ a.s. contains a Hamiltonian cycle. To show this result, Pósa used the extension-rotation technique. His result was an existence result, and he gave no algorithm to find a Hamiltonian cycle.

The threshold for Hamiltonicity was settled when Komlós and Szemerédi proved in [16] that the threshold for Hamiltonicity is equal to the threshold for minimum degree at least 2 (the latter threshold was proven by Erdôs and Rényi and was a known result). Denote these properties by 'HAM' and '$\delta \geq 2$', then their result was that for a random graph $G_{n,m}$ with $m = \frac{n}{2}(\log n + \log\log n + c_n)$ the following hold:

$$\lim_{n\to\infty} \mathbb{P}[G_{n,m} \in \text{'HAM'}] = \lim_{n\to\infty} \mathbb{P}[G_{n,m} \in \text{'}\delta \geq 2\text{'}] =$$
$$= \begin{cases} 0 & \text{if } c_n \to -\infty \\ e^{-e^{-c}} & \text{if } c_n \to c \\ 1 & \text{if } c_n \to \infty \end{cases}$$

We observe that their threshold is a sharp one. Moreover, since 'HAM' $\subseteq$ '$\delta \geq 2$', then in order to prove the threshold, it is sufficient to prove that a.e. graph with minimum degree 2 is Hamiltonian.

2.2.3 Hitting Times

The result of the previous section can be described in a stronger way using the notion of graph process and hitting time. A graph process $\tilde{G}$ is a nested sequence of graphs, $G_0(V, E_0) \subseteq G_1(V, E_1) \subseteq \ldots \subseteq G_N(V, E_N)$, where $E_0 = \emptyset$, and $E_i = E_{i-1} \cup \{e_i\}$, $e_i \in \overline{E}_{i-1}$. That is, a graph process is a sequence of nested graphs, which begins with the empty graph and which ends with the complete graph.

Since each permutation of the N edges gives rise to a graph process, we have $N!$ graph processes (for a given n). To turn the set of graph processes into a probability space, we take all graph processes to be equiprobable.

For a graph process $\tilde{G}$ and an increasing property Q, we define the hitting time of Q:

$$\tau(\tilde{G}, Q) = \min\{i : G_i \in Q\}$$

Clearly, for any graph process $\tilde{G}$,

$$\tau(\tilde{G}, \text{'}\delta \geq 2\text{'}) \leq \tau(\tilde{G}, \text{'HAM'})$$

In [1] and in [4] it was proven that

$$\tau(\tilde{G}, \text{'}\delta \geq 2\text{'}) = \tau(\tilde{G}, \text{'HAM'}), \text{ for a.e. graph process } \tilde{G}.$$

2.2.4 Algorithms for Finding Hamiltonian Cycles

Several algorithms were suggested to find Hamiltonian cycles in sparse random graphs. Most of these algorithm used the extension-rotation technique.

In [5] Bollobás , Fenner and Frieze suggested the algorithm HAM. They proved that for a random graph $G_{n,m}$, the probability that HAM finds a Hamiltonian cycle in $G_{n,m}$ is asymptotically equal to the probability that $G_{n,m}$ is Hamiltonian. The algorithm we present in this work is an adaptation of the HAM algorithm for the semi-random case. A discussion of the HAM algorithm is given in Chapter 4.

Several algorithms were suggested for the dense case. In [20], Thomason suggested an algorithm which works for $p(n) \geq 12n^{-1/3}$, and has expected linear time.

Chapter 3

The Semi-Random Model

3.1 Introduction

A semi random model combines two elements. One element is an adversary, and the other element is randomness. Both elements are combined to create the instances of the model. For example, many semi-random models have the following two stages structure. In the first stage a random graph is selected according to some random graph model, such that the generated graph has a planted solution or has a solution whp. Then, in the second stage, an adversary may modify the graph according to some restrictions.

The advantage of using the semi-random model is that it enables us to create instances which are "between" random instances and the worst-case instances. One reason to do so is that sometimes, finding a solution to a problem on a random graph is not very difficult. Moreover, algorithms for random graphs tend to rely on the statistical properties of random graphs, and therefore, these algorithms might perform badly on many other instances which do not have these properties. On the other extreme, we do not except to find algorithms for the worst-case instances, since these problems are usually NP-hard.

The specific details of the semi-random model depend on the investigated problem. These details include, for example, the restrictions on the adversary and the random model to be used.

We continue with the details of the semi-random model that is used in this work. Then we present some semi-random models for other NP-hard problems. Finally, we discuss some papers that deal with the problem of Hamiltonicity on models which are partially randomized.

3.2 A Semi-Random Model for the Hamiltonicity Problem

The semi-random model we use has two stages:

1. A random graph $G_1(V, E_1)$ is randomly selected from a probability space, such that G_1 is a.s. Hamiltonian.

2. An adversary adds a set of edges E_2 to the graph G_1. The final problem instance is $G(V, E_1 \cup E_2)$

That is, we start with a random graph G_1 and add arbitrary edges to get the final input graph G. Since Hamiltonicity is an increasing property, then if G_1 is Hamiltonian, so is the final graph G.

When an algorithms receives a semi random graph as an input, it is not specified which edges are the random edges and which edges were added by the adversary. Otherwise, the algorithm can remove the edges that were added by the adversary and restore the random graph.

Remark 3.2.1 *The edges of the set E_1 are called* original *edges, and the edges $E_2 \backslash E_1$ (i.e. the edges the adversary adds), are called* fake *edges.*

Whenever referring to properties of a semi random graph, unless explicitly said otherwise, we refer to the properties of the random part. For example, for a vertex $v \in G$, $\deg(v)$ is the degree of the vertex in G_1 (the random part), and $\deg_{\text{semi}}(v)$ is the degree of the vertex in G.

3.3 Semi-Random Models for Other NP-Hard Problems

We present here semi-random models for three NP-hard problems. The first problem is k-coloring a graph. The second problem is finding a maximum independent set (MIS). The third problem is finding a minimum bisection in a graph. Even though all these problems are NP-hard, they have different nature than the problem of Hamiltonicity (for example, a solution might not be preserved when adding arbitrary edges). Hence, a different random model is used here. The random model for these problems is the model of planted solution. In this model, we first create a solution for the discussed problem, then we add random edges with the restriction that the planted solution remains a solution.

In [10], Feige and Kilian proposed semi-random models for these problems, based on this random model. This semi-random model has three stages:

Stage 1 In this stage a solution to the problem is created. This is done by randomly selecting the sets of vertices to form the solution. That is, for the MIS problem, a set S of αn vertices is randomly chosen. For the problem of k-coloring, the set of vertices is randomly partitioned into k sets of size $\frac{n}{k}$. For the bisection problem, a set S of half of the vertices is randomly chosen.

Stage 2 In this stage we add random edges to the graph with the restriction that the planted solution remains a solution. That is, for the MIS problem, each edge between S and $\overline{S}$ is included with probability p. For the

k-coloring problem, each edge connecting different sets is included with probability p. For the bisection problem, each edge between S and $\overline{S}$ is included with probability q, and each other edge is included with probability p, where $p > q$. p and q (q is relevant only for the last problem) are the parameters of the random models.

Stage 3 In this stage the adversary modifies the graph, again with the restriction that the planted solution remains a solution. That is, for the MIS problem, it can add any edges which are not in $S \times S$. For the k-coloring problem the adversary can add edges that are between sets. For the bisection problem, it can remove any edges from $S \times \overline{S}$, and add any edges that are not between $S \times \overline{S}$.

For all the semi-random models, the question is whether the algorithm can reveal a solution which is either the planted one or as good as the planted one. The adversary has more freedom when the random parameters are lower. In the extreme case, taking the random parameters to be 0 enables the adversary to generate worst-case instances. Hence, the performance of the algorithms depends on the random parameters.

Feige and Kilian showed algorithms for these problems. For the MIS and for the k-coloring problem they prove that their algorithms are close to optimal for the semi-random models. For the MIS problem they proved that if $p = (1+\epsilon)\log n/\alpha n$, for any positive constant ϵ, their algorithm succeeds whp to recover an independent set of size αn. On the other hand, they proved that unless NP $\subseteq$ BPP, then if $p < (1-\epsilon)\log n/n$ for some $\epsilon > 0$, then any algorithm would fail whp against an optimal adversary. They used their algorithm for the MIS problem to obtain an algorithm for the k-coloring problem. For the bisection problem they proved that when the difference $p - q$ is high enough, their algorithm succeeds whp.

The problem of k-coloring was discussed earlier by Blum and Spencer in [2]. The semi-random models which were used there were of "noisy" adversary. In these models an adversary partitioned the vertices into k sets (in one model, there was restriction that each set must be of size $\Omega(\frac{n}{k})$, and in another model, there was no restriction on the size of the sets). Then for each edge which connects vertices from different sets, the adversary decided whether or not to include it in the graph. After each decision, with probability p the decision was reversed, hence p is called the noise rate.

3.4 Partially Random Models for Hamiltonicity

We discuss here two papers that discuss models of graphs, which are also only partially randomized.

In [6], Broder, Frieze and Shamir suggested the following model: start with a cycle in length n and add dn random edges, where d is a constant. They showed an algorithm, that if d is large enough, finds a Hamiltonian cycle whp in polynomial time (but which may not be the original one). This model resembles the random model of planted solution discussed in section 3.3. The motivation here was answering the question, whether adding a small number of random edges makes it hard to find a Hamiltonian cycle. A positive answer could lead to an authentication scheme (given graph G, the authentication is the Hamiltonian cycle H), but since they devised an algorithm to find a Hamiltonian cycle, this was not the case.

Another partially random model, was discussed in [3] by Bohman, Frieze and Martin. Their approach was a bit different. Given an arbitrary graph, from some class of graphs, the question they asked is how many random edges should be added in order to ensure Hamiltonicity whp. In their paper, the initial graph was an arbitrary graph with minimum degree dn, where $0 \leq d \leq \frac{1}{2}$. Their result was that it suffices to add to such a graph $100n\log(1/d)$ random edges in order to turn it to Hamiltonian whp. Their proof is constructive, but no time analysis was done to show it yields an efficient algorithm.

Chapter 4

The HAM Algorithm and the Semi-Random Model

In this chapter we discuss some of the problems that arise when running the algorithm HAM on semi-random graphs. These problems are later solved by the Semi-Ham algorithm, which is presented in Chapter 5.

We start by presenting the algorithm HAM and a sketch of its analysis. Then we move to discuss some problems HAM might have if the input is a semi random graph.

4.1 The HAM Algorithm

The HAM algorithm advances in stages. In each stage we try to extend the current path into a path which is one edge longer, which is the input for the next stage. In each stage the HAM algorithm uses the extension-rotation technique in the following way. We start with a path P, we then build a tree of paths, whose root is P. The tree is built level by level. The paths in level $t+1$ are all the paths generated by a single rotation from any path in level t. All rotations are allowed, but the depth of the tree is bounded. If during this process we succeed to extend any path, we directly move to the next stage. A pseudo code of a stage in the algorithm is given in Table 4.1.

4.2 Analysis of the Algorithm HAM

The main theorem about HAM is

Theorem 1 *Let* $m = m(n) = \frac{n}{2}(\log n + \log\log n + c_n)$. *Then,*

$$\lim_{n\to\infty} \mathbb{P}[\ \textit{HAM finds a Hamiltonian cycle in } G_{n,m}\ ] = \begin{cases} 0 & \textit{if } c_n \to -\infty \\ e^{-e^{-c}} & \textit{if } c_n \to c \\ 1 & \textit{if } c_n \to \infty \end{cases}$$

Algorithm HAM, Stage r

Input : Path P_r of length r, whose endpoints are v_1 and v_r

1. Insert P_r as the root of the tree.

2. t = 0 [current level of the tree]

3. while $t \leq 2T$ do $\left[T = \left\lceil \frac{\log n}{\log d - \log\log d} \right\rceil + 1,\ d \text{ is the average degree} \right]$

   a. For each path P in level t, call Expand-Path(P). If Expand-Path(P) returns a path of length $r+1$ (or a cycle of length n) then move to the next stage (or finish). Otherwise, add all the returned paths to the level $t+1$ in the tree.

   b. t = t + 1

Procedure Expand-Path

Input : Path P of length r, whose endpoints are w_1 and w_2

1. Let OUTPUT = $\emptyset$.

2. If either w_1 or w_2 has a neighbor not in P, then extend the path P using this edge. Return the extended path.

3. If there is an edge (w_1, w_2), then if $r = n$, return the Hamiltonian cycle, otherwise, return the path of length $r+1$ generated by a cycle extension.

4. For each $x \in N(w_1)$, add Rotate$(P, (x, w_1))$ to OUTPUT

5. For each $x \in N(w_2)$, add Rotate$(P, (x, w_2))$ to OUTPUT

6. return OUTPUT

Table 4.1: Algorithm Ham. Description of Stage r of the algorithm, and its subroutine Expand-Path

If $c_n \to -\infty$, then a.s. there is no Hamiltonian cycle, and hence HAM would fail, and we are done. Hence, assume $c_n \not\to -\infty$. Let $d = 2m/n$ be the average degree, and call a vertex v *small* if $\deg(v) \leq d/20$, and otherwise *large*. Then, the following claim was proven in [5, Lemma 3.1].

Claim 4.2.1 *Let $G_{n,m}$ be a random graph. Provided that $c_n \not\to -\infty$, the following a.s. hold.*

1. *$G_{n,m}$ contains no more than $n^{1/3}$ small vertices.*
2. *$G_{n,m}$ does not contain two small vertices at distance of 4 or less apart.*
3. *$G_{n,m}$ contains no vertex of degree exceeding $5d$.*
4. *For every set of large vertices S, with $|S| \leq n/d$, $|N(S)| \geq d|S|/300$.*

Let $m = m(n) = \frac{n}{2}(\log n + \log\log n + c_n)$ for $c_n \not\to -\infty$, and let Γ_0 be the set of all graphs with n vertices and m edges. Define

$$\Gamma_1 = \{G \in \Gamma_0, G \text{ is connected, } \delta(G) \geq 2, \text{ and } G \text{ satisfies Claim 4.2.1 } \},$$

and define

$$\Gamma_2 = \{G \in \Gamma_1, \text{ HAM failed on } G\}.$$

Almost every graph in Γ_0 is connected (this was proven, for example in [9]). Hence Claim 4.2.1, and the threshold for minimum degree 2 imply that,

$$|\Gamma_1| = (1 - o(1))e^{-e^{-c}}|\Gamma_0|.$$

Hence in order to complete the proof of Theorem 1 it is enough to prove that $|\Gamma_1|/|\Gamma_0| \to 0$. They prove a stronger result

$$\lim_{n\to\infty} \frac{|\Gamma_2|}{|\Gamma_0|} = 0 \tag{4.1}$$

The outline of the proof of Eq. (4.1) is as follows. Consider a graph $G(V, E) \in \Gamma_2$, and let r be the stage on which HAM fails. Let P_r be the initial path to this stage. Let $P_1, P_2, \ldots, P_{r-1}$ be the initial paths to each of the stages before r. Observe that each P_i was generated from P_{i-1} by a sequence of rotations and an extension. Hence P_r is the final path in a sequence of paths $P_1 = P^0, P^1, \ldots, P^M = P_r$, where each P^i was generated from P^{i-1} by a simple extension, a cycle extension or a single rotation. Let $W(G) = \bigcup_{i=1}^M E(P^i)$.

Let G' be a graph generated from G by deleting a subset of $E \backslash W(G)$. Then HAM would also fail on G', moreover, the same initial path for stage r would be generated, via the same sequence of paths P^i. This observation leads to the definition of a *deletable* set.

Definition 4.2.2 *A set of edges X, is called a* deletable *set, if*

- *No edge of X meets small vertices.*
- *No large vertex meets more than $d/1000$ edges of X.*
- $X \cap W(G) = \emptyset$.

For a set of edges X, define $G_X = G(V, E \backslash X)$. Consider an execution of HAM on $G \in \Gamma_2$ which fails on stage r. Let v_1, v_r be the endpoints of the initial path. Recall that the depth of the tree built by HAM is bounded by $2T$. For a path Q in the tree, let depth(Q) be its depth. Define,

$$\mathrm{END}(G) = \{v : \exists \text{ path } Q \text{ whose endpoint is } v,\ \mathrm{depth}(Q) \le T\} \ .$$

For each $v \in \mathrm{END}(G)$, define,

$$\mathrm{END}(G, v) = \{w : \exists \text{ path } Q \text{ whose endpoints are } v \text{ and } w,\ \mathrm{depth}(Q) \le 2T\} \ .$$

Then, the following was proven in [5, Lemma 3.2],

Claim 4.2.3 *Let $G \in \Gamma_2$, and assume HAM failed on stage r. Then for any deletable set X,*

1. $|\mathrm{END}(G_X)| \ge \frac{n}{1000}$
2. $\forall v \in \mathrm{END}(G_X),\ |\mathrm{END}(G_X, v)| \ge \frac{n}{1000}$

Moreover, we observe that for any $v \in \mathrm{END}(G_X)$ there is no edge between v and $\mathrm{END}(G_X, v)$, since otherwise, a cycle extension could be done.

To finish the proof of Eq. (4.1) a counting argument was used. Let $w = \lceil \lambda d \rceil$, for some constant $\lambda > 0$. For any graph $G \in \Gamma_0$, and for any set X of w edges, define $a(G, X) = 1$ if

- HAM failed on G and on G_X.
- X does not contain an edge (v, w), s.t. $v \in \mathrm{END}(G_X)$ and $w \in \mathrm{END}(G_X, v)$
- $|\mathrm{END}(G_X)| \ge \frac{n}{1000}$, and for all $v \in \mathrm{END}(G_X, v)$, $|\mathrm{END}(G_X, v)| \ge \frac{n}{1000}$

and otherwise $a(G, X) = 0$.

Observe that if $G \in \Gamma_2$ and X is deletable set, then $a(G, X) = 1$. Hence

$$\sum_{G \in \Gamma_2} \overbrace{|\{X \subseteq E(G) : |X| = w, X \text{ is deletable}\}|}^{(a)} \le \overbrace{\sum_{G \in \Gamma_0} \sum_{\substack{X \subseteq E(G) \\ |X| = w}} a(G, X)}^{(b)}$$

Then, they gave a lower bound for (a), by showing that almost any choice of w edges, would yield a deletable set. On the other hand, they gave an upper bound for (b), and use these bounds to prove Eq. (4.1).

4.2.1 Time Complexity

To analyze the time complexity, we count the number of basic operations: a simple extension, a cycle extension and a rotation. In each stage, the number of basic operations is as the size of the tree. Since the depth of the tree is bounded by $2T$, and since the degree of each vertex is at most $5d$, then the size of the tree is polynomial, and so is the running time.

4.3 Problems on Semi-Random Inputs

In this section we discuss some of the problems that can occur if the input to HAM is a semi random graph instead of a random graph. In general, given an algorithm for random graphs, the following may occur when the input is semi-random.

1. The algorithm and its analysis are valid for the semi random case.
2. The algorithm remains valid, but its analysis needs to be fixed.
3. The main concepts and techniques of the algorithm can be used. However, some modifications have to be done both in the algorithm and in its analysis.
4. The algorithm cannot be used in the semi random case, and a new approach has to be taken in order to solve the problem.

For example, the algorithm that Thomason suggested in [20] for finding Hamiltonian cycles in dense random graphs (see Section 2.2.4) works with no changes in the semi-random case.

On the other hand, the problems discussed in Section 3.3 were solved by Feige and Kilian [10] using semi-definite programming, which is more elaborate tool than eigenvalue analysis, which was used to achieve the previously best results for these problems.

For our problem, the basic concepts of HAM can be used, but some changes have to be done in the algorithm and in its analysis in order to use them in the semi-random case. We discuss here some of the reasons for that.

4.3.1 Running time

The first problem that prevents us from using the algorithm as it is, is the running time. Recall that in HAM the running time depends on the size of the tree, which itself is polynomial since the depth of the tree was bounded and the maximum degree was $5d$. In semi random graphs, the maximum degree is no longer bounded, and the tree may no longer be of polynomial size.

In order to solve this problem, the size of the tree has to be bounded. One way to do this, is to ignore some of the paths in the tree, for example, in Semi-Ham, we ignore paths whose endpoints already appeared in the tree.

4.3.2 Distance Between Small Vertices

Property 2 in Claim 4.2.1 states that the distance between small vertices is at least 5. This property is crucial in analyzing the growth rate of the tree. The idea is as follows. Assume the current stage is r, and that in level t we have found a path $P = (v_1, v_2, \ldots, v_{r-1}, v_r)$. Consider v_r, let $v_{i_1}, \ldots, v_{i_k}$ be its neighbors. Then for each neighbor v_{i_j}, we can rotate with the edge (v_r, v_{i_j}) as the rotation edge. As a result of this rotation we would get a path with v_{i_j+1} as an endpoint.

If v_r is large then at most one of the generated endpoints is small, and if v_r is small then none of the generated endpoints is small. To see that, if v_r is large and v_{i_j+1} and v_{i_k+1} are small, then there is a path of length 4, $(v_{i_j+1}, v_{i_j}, v_r, v_{i_k}, v_{i_k+1})$ between small vertices. If v_r and v_{i_j+1} are small, then there is a path $(v_r, v_{i_j}, v_{i_j+1})$ of length 2 between small vertices. This fact and property 4 in Claim 4.2.1 are combined to prove that the ratio of the size of two consecutive levels in the tree is $\Omega(\log n)$, which is then used to prove Claim 4.2.3.

Since this property is no longer valid, then the ratio may be smaller than $\Omega(\log n)$, and as a result, the tree may not be big enough to satisfy Claim 4.2.3.

To illustrate this, consider the initial path P_r shown in Figure 4.1. Its endpoint v_r is a small vertex, but since Property 2 is no longer valid, then in the first level of the tree all the new endpoints are small (For P_r it is v_{i+1},v_{j+1} and v_{k+1}). This situation can repeat for any of the paths in the first level, and so on.

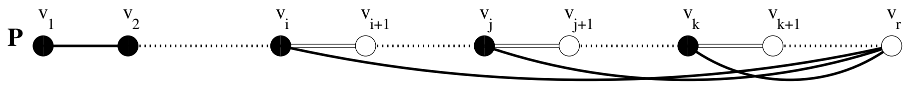

Figure 4.1: The path P_r whose endpoint is v_r. White vertices are small, fake edges are depicted as double edges. After rotating, all the new endpoints are small

On the extreme case, if all endpoints are of degree 2, then in each level there would be only one new path. In this case, assuming that p is close to the threshold, then since the expected number of vertices of degree 2 is $\Omega(\log n)$, the tree would be a chain.

In order to solve this problem, both the algorithm and its analysis are changed.

For the analysis, we refine the partition of the vertices by their degrees, this enables us to do a better analysis of the growth rate, according to the vertices that are already in the tree.

In order to handle the situations in which the growth rate is smaller than $\Omega(\log n)$, we allow a deeper tree. Yet, this does not help us to solve the problem of vertices of degree 2 (the case of the chain that was discussed above), and we add further changes to the algorithm to handle this case.

4.3.3 Consequences of Restricting the Rotations

As discussed in Section 4.3.1, in order to keep the size of the tree polynomial, we need to restrict some of the rotations. For example, we keep at most one path for any pair of endpoints. In this section, we discuss some of the consequences this may have.

Consider two graphs G and G' where G' was generated from G by adding edges. If both graphs have the same initial path for stage r of HAM, then in stage r, the tree that HAM builds for G is contained in the tree that HAM builds for G'. This is since all rotations are allowed. But, since Semi-Ham ignores some of the paths, then as a result, the tree of G may not be contained in the tree of G'.

For example, let G and G' be such that G' was generated from G by adding the edge (v_r, v_{j-1}) (which is missing in G). Consider the situation in Figure 4.2.

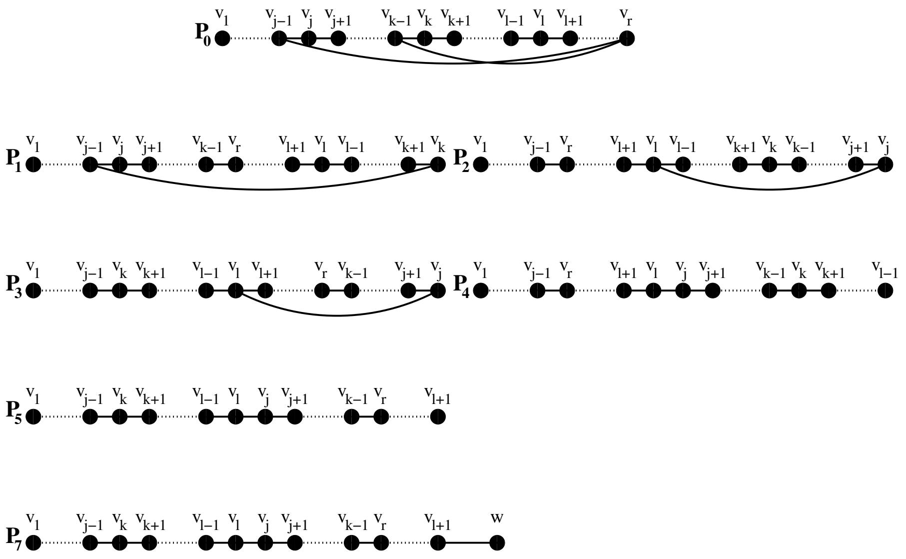

Figure 4.2: The tree HAM builds for the graph G', the final path is P_7. The tree HAM builds for G is the left side of the tree, again the final path is P_7

During the original execution of HAM both on G' and on G, the final path would be P_7. Consider now an execution of HAM where we ignore paths whose endpoints already appeared. In this case, if the input graph is G then the final path is P_7, but if the input is G' then the path P_3 is ignored (since P_2 is already a path whose endpoints are v_1 and v_j) and the final path in this case is not P_7 and we may even fail.

As we have seen, if not all rotations are allowed, then adding edges may cause us to avoid solutions. The adversary can take advantage of this idea, to lead us in the wrong way.

Another consequence of restricting the rotations is that the counting argument used in the analysis of HAM is no longer valid. The reason is that the basic idea in the counting argument was that for a deletable set X, if we failed on G then we would also fail on G_X, but as was shown, we may succeed on G_X and fail on G.

Chapter 5

The Semi-Ham Algorithm

5.1 Introduction

In this chapter we present the algorithm Semi-Ham, which is an adaptation of the algorithm HAM for the semi random case. Our main theorem is the following.

Theorem 2 *Let G be a semi random graph which is generated by adding arbitrary edges to a random graph $G_{n,m}$, with $m(n) = \frac{n}{2}(\log n + \log\log n + c_n)$, where $c_n \nrightarrow -\infty$. Then*

$$\lim_{n\to\infty} \mathbb{P}[\textit{Semi-Ham finds a Hamiltonian cycle in } G\ ] = \begin{cases} e^{-e^{-c}} & \textit{if } c_n \to c \\ 1 & \textit{if } c_n \to \infty \end{cases},$$

where probability is taken over the choices of $G_{n,m}$.

That is, for almost every random graph $G_{n,m}$, no matter what the arbitrary edges that we add, Semi-Ham will find a Hamiltonian cycle.

The main changes in Semi-Ham in comparison to HAM are as follows.

- In each stage, not all the rotations are allowed. Restrictions are given on the allowed rotations, in order to ensure that the size of the tree of paths is polynomial.
- In each stage in HAM, one tree of paths is built, and its depth is $2T$. In Semi-Ham we split the tree into sub-trees. In each stage, Semi-Ham starts with building a tree of depth t_0, then each node in this tree is used as a root for another tree of depth t_0. This change is mainly technical, and it is done only to simply the presentation of the algorithm and of its analysis.
- As was just mentioned, the depth of the tree in HAM was $2T$ which is $\Theta(\frac{\log n}{\log\log n})$. In Semi-Ham the total depth is $2t_0 = \Theta(\frac{\log n}{\log\log\log n})$.
  Hence, in Semi-Ham the maximal number of rotations used to generate any path from the initial path is higher, and this is done since the growth rate of the tree is smaller.

Algorithm Semi-Ham

Input : A graph $G(V, E)$, $|V| = n$

1. Let w be an arbitrary vertex in G.
2. Let $P_1 = w$.
3. For each $i = 1, \ldots, n$
   a. Call the procedure Semi-Ham-Stage, with P_i.
   b. If the procedure succeeded, and $i = n$, then terminate and output the Hamiltonian cycle.
   c. If the procedure succeeded, and $i < n$, then let P_{i+1} be the returned path.
   d. Otherwise, (the procedure failed) terminate with failure.

Table 5.1: Algorithm Semi-Ham

- In HAM, in each stage we build the tree, and a.s. one of the paths in the tree can be extended. In Semi-Ham, we add another procedure which tries to close each of the paths in the tree into a cycle.

  This is done in order to use a different proof technique than was used for HAM. Instead of using the counting argument based on deletable sets, we use a proof technique taken from [1] and [17].

- In spite all of the changes in Semi-Ham, it still may fail in the cases where the random part of the input graph has vertices of degree 2. To handle this case, extra work is done in the beginning of each stage. This is described in Chapter 7.

5.2 General Description of Semi-Ham

Like the HAM algorithm, the algorithm Semi-Ham advances in stages. The first stage starts with a path of one vertex, then in each stage we try to extend the current path, into a path which has one more edge. If we succeed, then this path is the input for the next stage. The last stage starts with a path of length n, and terminates with a Hamiltonian cycle. A pseudo code of the main body of the algorithm is given in Table 5.1. The Semi-Ham algorithm uses the extension-rotation technique with the three basic path operations: simple extension, cycle extension and rotation, which were described in Section 2.2.1.

Each stage in the algorithm has two parts. The main procedure in the first part is Build-Tree. In the second part, the main procedure is Build-Tree-2. We first describe these procedures, then we describe the stage.

As mentioned above, the algorithm presented here is for inputs whose random part has minimum degree 3. In order to use the algorithm for the general case (that is, when the minimum degree is 2), some extra work has to be done. This is described in Chapter 7.

5.3 Basic Procedures

5.3.1 The Procedure Build-Tree

The motivation of the procedure Build-Tree is as follows. Given an initial path P_{v_1,v_r} of length r, we want to construct a set of paths of length r, all with v_1 as an endpoint, and all have the same set of vertices.

This is done in similar way to HAM. We build a tree of paths level by level. The root of the tree is the initial path. To build the next level from the current level, we go over all the paths in the current level. For each path, we first try to extend it by a simple extension or a cycle extension. If we could not extend, then we do all *allowed* (explained below) rotations, and add the new paths to the next level.

Unlike the situation in HAM algorithm, in Semi-Ham not every rotation is allowed. A rotation is *allowed* only if,

1. The broken edge is an edge of the initial path.
2. The new endpoint (of the path generated in the rotation) is not an endpoint of any path currently in the tree.

If we find $\frac{n}{\mu}$ paths (where $\mu = 1000$) then we stop building the tree. In this case, the algorithm succeeds. On the other hand, if we run out of allowed rotations, or if the depth of the tree is more than $t_0 = \frac{\log n}{c_1 \log \log \log n}$, for some constant c_1, then we fail. A pseudo code of the procedure Build-Tree is given in Table 5.2.

5.3.2 The Procedure Build-Tree-2

The procedure Build-Tree-2 is very similar to the procedure Build-Tree. The main difference is that in Build-Tree-2 only rotations are done, that is, the procedure does not not try to extend any path. The reason for that is explained in Section 5.4.2. The other difference is the stopping conditions. The procedure succeeds if $\frac{3n \log \log n}{\log n}$ paths are found, and fails if we exceed the depth of $\log n$, or if we run out of allowed rotations. A pseudo code of Build-Tree-2 is given in Table 5.3

5.4 The Procedure Semi-Ham-Stage

We turn now to describe a stage in the algorithm. The r-th stage of the algorithm starts with a path of length r, and unless it fails, terminates with a path of length $r + 1$.

Procedure Build-Tree

Input : Path P_{v_1,v_r} of length r, with v_1 as the fixed endpoint.

1. Insert P_{v_1,v_r} as the root of the tree.
2. t = 0 [current depth of the tree]
3. while $t \leq t_0$ do $\left[\ t_0 = \left\lceil \frac{\log n}{c_1 \log\log\log n} \right\rceil\ \right]$
   a. If the tree contains $\frac{n}{\mu}$ paths then return the tree. [success]
   b. If level t is empty, then fail. [we run out of allowed rotations]
   c. For each path P in level t,
      i. Call Expand-Path(P).
      ii. If the call returns a path of length $r+1$, or a a cycle of length n, then move to the next stage. Otherwise, add all the returned paths to level $t+1$ of the tree.
   d. t = t + 1.

Procedure Expand-Path

Input : Path $P_{v_1,w}$ of length r. v_1 is the fixed endpoint.

1. Let OUTPUT = $\emptyset$
2. If w has a neighbor $x \notin P_{v_1,w}$, then return the path $P \circ (w,x)$
3. If v_1 is a neighbor of w, then if $r = n$, return the Hamiltonian cycle, otherwise, return the path of length $r+1$ generated by a cycle extension.
4. For each $x \in N(w)$
   a. Let x' be the neighbor of x on $P_{v_1,w}$ which is closer to w.
   b. If there is no path in the tree with x' as an endpoint, and if the broken edge (x,x') is an edge of the initial path, then add Rotate($P_{v_1,w},(x,w)$) to OUTPUT.
5. Return OUTPUT

Table 5.2: The Procedure Build-Tree, and its subroutine Expand-Path

Procedure Build-Tree-2

Input : Path $P_{v,x}$, with x as the fixed endpoint.

1. Insert $P_{v,x}$ as the root of the tree.
2. t = 0 [current depth of the tree]
3. while $t \leq t_0$ do [$t_0 = \log n$]
   a. If the tree contains $\frac{n}{\mu}$ paths then return the tree. [success]
   b. If level t is empty, then fail. [we run out of allowed rotations]
   c. For each path P in level t
      i. Call Expand-Path-2(P).
      ii. Add all the returned paths to level $t+1$ of the tree.
   d. t = t + 1

Procedure Expand-Path-2

Input : Path $P_{w,x}$. x is the fixed endpoint.

1. Let OUTPUT = $\emptyset$
2. For each $y \in N(w)$
   a. Let y' be the neighbor of y on P which is closer to w.
   b. If there is no path in the tree with y' as an endpoint, and if the broken edge (y, y') is an edge of the initial path, then add Rotate$(P_{w,x}, (y, w))$ to OUTPUT.
3. Return OUTPUT

Table 5.3: The Procedure Build-Tree-2, and its subroutine Expand-Path-2

Every stage has two parts : Semi-Ham-Stage$_1$ and Semi-Ham-Stage$_2$. In Semi-Ham-Stage$_1$ we create a set of $\Theta(n^2)$ paths, all of them of length r. We will prove that among the paths found in first stage, there is at least one path, that we can divide into two sub-paths, and then by rotating each side of the path independently, we can close this path into a cycle. Hence, in the second part, Semi-Ham-Stage$_2$ we iterate over all the paths, and over all the intermediate points, and find this path. Details follow.

5.4.1 Semi-Ham-Stage$_1$

Let r be the current stage. Semi-Ham-Stage$_1$ starts with a path P_r of length r. Its aim is to find a set of $\Theta(n^2)$ paths, all have the same length and the same set of vertices. We fix the endpoint v_1, then we call Build-Tree to get a set of $\Theta(n)$ paths, all have v_1 as their endpoint. Then for each such path, we call again for Build-Tree, but this time with the other endpoint (i.e., which is not v_1) as the fixed endpoint, hence all together we have $\Theta(n^2)$ paths. Observe that in this case, any of these paths was generated by at most $2t_0$ rotations from the initial path P_r.

This part may not be completed. If any call to Build-Tree fails, then the algorithm fails. On the other hand, if any call returns a path of length $r+1$, then we move to the next stage (i.e, we skip also Semi-Ham-Stage$_2$). A pseudo code of the Semi-Ham-Stage$_1$ is given in Table 5.4.

5.4.2 Semi-Ham-Stage$_2$

The motivation of the second part is as follows. Let $P_{a,b}$ be a path, and let x be an intermediate point in the path. x divides $P_{a,b}$ into two edges disjoint sub-paths, the left one is $P_{a,x}$ and the right one is $P_{x,b}$.

By rotating each of these paths, using x as the fixed endpoint, we obtain two sets of endpoints: L and R. For each $l \in L$, there is a path $P_{l,x}$ of the same length of $P_{a,x}$, and with the same set of vertices. The same holds for each $r \in R$, with respect to the path $P_{x,b}$.

Since these rotations are done independently for each sub path, then for each $l \in L$ and $r \in R$, we can concatenate the paths $P_{l,x}$ and $P_{x,r}$ to obtain a path $P_{l,r}$ of the same length of $P_{a,b}$ and with same set of vertices. Moreover, if there is any edge (l,r) with $l \in L$ and $r \in R$, then we can close the path $P_{l,r}$ into a cycle and perform a cycle extension.

The building of the sets L and R is done by calling the procedure Build-Tree-2, since we are not trying to extend the partial paths here, only to rotate them.

We will prove that among all the paths found in Semi-Ham-Stage$_1$, there is at least one pair (a,b), who has an intermediate point x, for which the process described above succeed to close a cycle. Since we only prove the existence of such path, then in the second part, we iterate over all the paths found in Semi-Ham-Stage$_1$, and over all the intermediate points, and try each possibility. Pseudo code of Semi-Ham-Stage$_2$ is given in Table 5.5.

Semi-Ham-Stage$_1$

Input : A path P_r of length r, whose endpoints are v_1 and v_r.

1. Call the procedure Build-Tree with P_r as the initial path, and v_1 as the fixed endpoint.
2. If the procedure returns a path of length $r+1$ or a cycle of length n, move to the next stage.
3. If the procedure fails, then the algorithm fails.
4. Otherwise, let A be the set of endpoints of the returned paths.
5. For each vertex in $a \in A$

    a. Call the procedure Build-Tree with $P_{v_1,a}$ as the initial path, and a as the fixed endpoint.

    b. If the procedure returns a path of length $r+1$, or a cycle of length n move to the next stage.

    c. If the procedure fails, then the algorithm fails

    d. Otherwise, let $B(a)$ be the set of endpoints of the returned paths , which are not a.

Table 5.4: Semi-Ham-Stage$_1$, the first part of Semi-Ham-Stage.

Semi-Ham-Stage$_2$

Input : A set A of points, and for each $a \in A$, a set $B(a)$ of points, such that for each $a \in A$ and $b \in B(a)$ there is a path $P_{a,b}$ found in Semi-Ham-Stage$_1$.

1. For each pair (a, b), such that $a \in A$, $b \in B(a)$,

    a. Let $P_{a,b}$ be the path of length r with endpoints a and b

    b. For each intermediate vertex x on the path $P_{a,b}$,

        i. Let $P_{a,x}$ and $P_{x,b}$ be the two sub-paths.

        ii. Let L be the set of other endpoints obtained by calling to Build-Tree-2 with $P_{a,x}$, and with x as the fixed endpoint.

        iii. Let R be the set of other endpoints obtained by calling to Build-Tree-2 with $P_{x,b}$, and with x as the fixed endpoint.

        iv. If there is an edge (l, r), where $l \in L$ and $r \in R$, then return the path generated by a cycle extension from the cycle $P_{r,x} \circ P_{x,l} \circ (l, r)$.

Table 5.5: Semi-Ham-Stage$_2$, the second part of Semi-Ham-Stage

5.5 Time Complexity of Semi-Ham

We define a *trial* as an event in which we try to extend or rotate a path. We analyze the time complexity of Semi-Ham in terms of trials.

In each call to Build-Tree and Build-Tree-2, the size of the tree of paths is $O(n)$. When expanding a node, the number of trials that are done is $O(n)$, and hence the total number of trials during each call to Build-Tree or Build-Tree-2 is $O(n^2)$.

In Semi-Ham-Stage$_1$ $O(n)$ trees are built, and in Semi-Ham-Stage$_2$ $O(n^3)$ trees are built. Hence the complexity of Semi-Ham-Stage$_1$ is $O(n^3)$ and the complexity of Semi-Ham-Stage$_2$ is $O(n^5)$. There are n stages, and hence the time complexity of Semi-Ham is $O(n^6)$.

This analysis is an upper bound for the real running time. Since each stage is ended when we succeed to extend a path, then at least in the early stages, we expect to succeed in the beginning of the stage.

5.5.1 Randomization of Semi-Ham

In Semi-Ham-Stage$_2$ we iterate over $O(n^2)$ paths, and for each path we iterate over $O(n)$ intermediate points. These iterations are done in an arbitrary order. Instead, we can iterate in random order, that is, each time we pick at random a triple a (a, b, x), where $a \in A$, $b \in B(a)$ and $x \in P_{a,b}$.

The number of triples is $\Theta(n^3)$ triples. As we will show in Section 6.6, the number of triples for which we succeed to close a cycle is $\Omega(\frac{n^3(\log\log n)^5}{(\log n)^3})$. Hence, using a random approach, the probability that we close a cycle is at least

$$\Omega\left(\frac{(\log\log n)^5}{(\log n)^3}\right) .$$

Hence, the expected running of Semi-Ham-Stage$_2$ is $O(\text{polylog}(n) \cdot n^2)$, and in this case the dominant part is Semi-Ham-Stage$_1$, and therefore the expected running of Semi-Ham is $O(n^4)$ (in terms of trials).

Chapter 6

Proof of Theorem 2

6.1 Introduction

To prove Theorem 2, we use the following strategy. We first give a list of graph properties that almost every random graph $G_{n,m}$ with $m = m(n) = \frac{n}{2}(\log n + \log\log n + c_n)$ $c_n \nrightarrow -\infty$, has. Next we prove that for a semi-random graph G, if its random part has these properties and has minimum degree 2, then Semi-Ham will find a Hamiltonian cycle in G.

We assume in this chapter, that the minimum degree is 3. In fact, this assumption is used only once, see Remark 6.4.4. In Chapter 7, we eliminate this assumption, and complete the proof for the general case.

Whenever it is needed, we assume that n is large enough, such that all approximations are valid.

6.2 Properties of Random Graphs

We begin the proof with a list of graph properties, which are divided into four categories. In all the properties below, assume that $G(V, E)$ is a graph, with $|V| = n, |E| = m$.

6.2.1 Degree of Vertices

The first category is of properties related to the degrees of vertices in a graph. Let c_2 be an arbitrary large constant. We partition the vertices in the graph into four categories:

$$\text{LARGE} = \left\{ v : \deg(v) > \tfrac{\log n}{20} \right\}$$

$$\text{SMALL} = \left\{ v : (\log\log n)^{c_2} \leq \deg(v) \leq \tfrac{\log n}{20} \right\}$$

$$\text{TINY} = \{ v : 2 \leq \deg(v) < (\log\log n)^{c_2} \}$$

In addition, we define the set

$$\text{TWO} = \{v : \deg(v) \leq 2\} ,$$

which is a subset of TINY.

The definition of LARGE is motivated from the fact that the graphs that we are interested in have average degree of at least $\log n$. As discussed in Section 4.3.2, the partition of the vertices into large and non-large is not sufficient for the analysis, and therefore we add the set TINY. The exact value of c_2 can be determined from the rest of the proof.

Using this partition, we define the following properties.

P1. $|\text{LARGE}| \geq n - n^{1/3}$

P2. $|\text{TINY}| \leq e^{\frac{\log n}{\log\log n}} = n^{\frac{1}{\log\log n}}$

P3. $|\text{TWO}| \leq \frac{(\log n)^2}{c_3}$, for arbitrary large constant c_3.

Note that properties P1-P3 are increasing properties.

6.2.2 Increasing Properties

In the second category we give a list of graph properties which are preserved when adding edges.

P4. For any two disjoint sets $A, B \subseteq V$, s.t. $|A|, |B| \geq \frac{3n \log\log n}{\log n}$, there is an edge between A and B.

P5. For any two disjoint sets $A, B \subseteq V$, s.t. $|A|, |B| \geq \frac{20n \log\log n}{\log n}$, the number of edges between A and B is at least $\frac{|A||B| \log n}{4n}$.

P6. G is connected.

6.2.3 Expansion Properties

In this section we give properties related to the size of the neighborhood of sets of vertices.

P7. For each set of large vertices $S \subseteq V$, with $|S| \leq \frac{n}{\log n}$, $|N(S)| \geq \frac{\log n}{300}|S|$

P8. For each set of small vertices S , $|N(S)| \geq (\log\log n)^{c_2 - 1}|S|$.

P9. Each set of tiny vertices S with $|S| \geq \log\log n$,

- if the minimum degree in S is 3 then $|N(S)| \geq \frac{9}{4}|S|$.
- if the minimum degree is 2, but at most $|S|/4$ of the vertices are of degree 2 then $|N(S)| \geq \frac{9}{4}|S|$.

P10. For any set S, $|S| \leq \frac{n}{\log n}$, if the minimum degree in S is 2 then $|N(S)| \geq 2|S|$.

These properties are not increasing. However, for our case consider a random graph $G_{n,m}$ that satisfies these properties, and a semi-random graph G whose random part is $G_{n,m}$. Then G satisfies these properties when the conditions on the degrees are with respect to the degrees on $G_{n,m}$ (see Remark 3.2.1).

6.2.4 Decreasing Properties

The last property is preserved when deleting edges.

P11. For any two disjoint sets $A, B \subseteq V$, s.t. $|A|, |B| \leq \frac{30n \log \log n}{\log n}$, the number of edges between A and B is at most $300(\log \log n) \max\{|A|, |B|\}$.

6.2.5 Existence of Properties on Random Graphs and on Semi-Random Graphs

Consider a random graph $G_{n,m}$, with $m(n) = \frac{n}{2}(\log n + \log \log n + c_n)$, and $c_n \not\to -\infty$. We want to use the properties listed above in order to prove that Semi-Ham finds a Hamiltonian cycle whp in a semi-random graph G, whose random part is $G_{n,m}$. Since P1-P6 are increasing, and P7-P10 are preserved when the parameters are taken w.r.t to the random graph (as explained in Section 6.2.3), then if $G_{n,m}$ has these properties, then so does G.

The problem is with the decreasing property. Not only that it is not preserved when adding arbitrary edges, it does not hold when the number of edges is high (for example, if $m(n) = \Omega(n^2)$). In order to use these properties in our analysis, we claim that $G_{n,m}$ contains a spanning subgraph that satisfies the decreasing property, and some of the increasing properties (without this requirement, we could take a graph with no edges as the spanning subgraph, which would give us nothing). Since this spanning subgraph is contained in $G_{n,m}$ then it is contained in the semi-random graph G. Details follow.

For properties P1-P10 we would prove in Appendix A, that if $m(n) = \frac{n}{2}(\log n + \log \log n + c_n)$, and $c_n \not\to -\infty$, then a.s. $G_{n,m}$ has these properties.

Let $m' = m'(n) = \frac{n-1}{2} \log n$. In fact, for properties P4 and P5, we prove that they a.s. hold if $m(n) \geq m'$. For the decreasing property, P11, we prove that it a.s. holds if $m(n) \leq m'$. Hence, if $G_{n,m}$ is a random graph with $m = m'$, then $G_{n,m}$ a.s. has properties P4, P5 and P11.

Consider a random graph $G_{n,m}$ with $m \geq m'$. Then $G_{n,m}$ can be seen as constructed in two phases. First, a random graph with m' edges were chosen, and then $m - m'$ edges were selected from the remaining $N - m'$ edges. This leads to the following claim.

Claim 6.2.1 *Let $G_{n,m}$ be a random graph with $m(n) > m'$. Then $G_{n,m}$ a.s. contains a spanning subgraph that satisfies properties P4, P5 and P11.*

We combine all the above to the following lemma.

Lemma 6.2.2 *Let $G_{n,m}$ be a random graph with $m(n) = \frac{n}{2}(\log n + \log \log n + c_n)$, $c_n \not\to -\infty$. Then $G_{n,m}$ a.s. satisfies properties P1-P10. Moreover, $G_{n,m}$ a.s. contains a spanning subgraph that satisfies properties P4, P5 and P11.*

Throughout this chapter we refer to the set of edges of the spanning graph as the set of *green* edges. Whenever we index some expression with *green*, then it is w.r.t. the green edges.

Finally, for a given $m = m(n) = \frac{n}{2}(\log n + \log\log n + c_n)$, $c_n \not\to -\infty$, we let Γ_0 be the set of all graphs with n vertices and m edges. Let

$$\Gamma_1 = \{G \in \Gamma_0 \ : \ G \text{ satisfies Lemma 6.2.2 and } \delta(G) \geq 2\}$$

6.3 Reduction of Theorem 2

Let G be a semi random graph. We claim that if the random part of G is in Γ_1, then this is a sufficient condition of the success of Semi-Ham, that is, no matter what the adversary adds to the graph.

Theorem 3 *Let G be a semi random graph, whose random part $G_{n,m}$ is in Γ_1. Then Semi-Ham finds a Hamiltonian cycle in G.*

Since almost every graph in Γ_0 satisfies Lemma 6.2.2, then

$$\lim_{n\to\infty} |\Gamma_1| = e^{-e^{-c}}|\Gamma_0| \ ,$$

and hence Theorem 3 implies Theorem 2, and would prove our main assertion about Semi-Ham.

In the rest of the proof we prove the correctness of Semi-Ham on input graphs whose random part is in Γ_1. In Section 6.4 we prove the correctness of Semi-Ham-Stage$_1$. That is, we prove that if the current stage is r, and the input graph G has its random part in Γ_1 then Semi-Ham either finds a path of length $r+1$, or it generates $\Theta(n^2)$ paths of length r.

In Section 6.5, we prove the correctness of Semi-Ham-Stage$_2$. That is, we prove that one of the paths generated in Semi-Ham-Stage$_1$, can be closed into a cycle. Then if $r < n$, we can perform a cycle rotation and move to the next stage, and if $r = n$, then a Hamiltonian cycle is found.

6.4 Proof of Semi-Ham-Stage$_1$

Theorem 4 *Let G be a semi-random graph, whose random part $G_{n,m} \in \Gamma_1$. Let r be the current stage of Semi-Ham, and let the sets A ,$B(a)$-s be defined as in the algorithm. If during Semi-Ham-Stage$_1$ no path of length $r+1$ was found, then in the end of Semi-Ham-Stage$_1$,*

- $|A| \geq \frac{n}{\mu}$.
- *For each $a \in A$, $|B(a)| \geq \frac{n}{\mu}$.*

Recall that for any pair a, b where $a \in A$, $b \in B(a)$, there is a path $P_{a,b}$ generated by at most $2t_0$ rotations from the initial path of stage r.

Consider a call to the procedure Build-Tree with an initial path P_{v_1,v_r} of length r, and with v_1 as the fixed endpoint. Assume no simple or cycle extension were done. Define,

$$\text{OTHER} = \{v : v \text{ is an endpoint of a path returned by Build-Tree}, v \neq v_1\}.$$

Then Theorem 4 is a simple corollary of the following claim.

Claim 6.4.1 *Let G be a semi-random graph, whose random part $G_{n,m} \in \Gamma_1$. Then in any call to Build-Tree, provided that no extension were done,*

$$|\text{OTHER}| \geq \frac{n}{\mu} .$$

Proof of Claim 6.4.1: Let $P_{v_1,v_r} = (v_1, v_2, \ldots, v_r)$ be the input path to Build-Tree. Let v_1 be the fixed endpoint. We denote by S_t the set OTHER up to depth t. Hence, for each $v \in S_t$ there is a path $P_{v_1,v}$ generated by **at most** t rotations from P_{v_1,v_r}, using only edges of P_{v_1,v_r} as broken edges. Our aim is to analyze the growth rate of the set S_t. We use a technique similar to one used in [17].

We define the following set

$$T = \{v_i \in N(S_t) \ : \ v_i, v_{i-1}, v_{i+1} \notin S_t\}.$$

The set T is subset of $N(S_t)$. Observe that a vertex is in T if its neighbors along the initial path P_{v_1,v_r} and itself are not in S_t. Since any vertex in S_t, except v_r, has two neighbors along the path, and v_r itself has one neighbor, then,

$$|T| \geq |N(S_t)| - (2\,|S_t| - 1) \ . \tag{6.1}$$

For any $v_i \in T$, there is an edge (v', v_i) where $v' \in S_t$. Moreover, since $v_i \in T$, then the edges (v_{i-1}, v_i) and (v_i, v_{i+1}) were not deleted during any of the rotations done up to depth t. This is since an edge is deleted only if one of its vertices becomes a new endpoint, and none of v_{i-1}, v_i and v_{i+1} is in S_t.

Let $P_{v_1,v'}$ be the path in the tree whose endpoint is v'. Then, using the arguments above, rotating $P_{v_1,v'}$ with (v', v_i) as the rotation edge is an allowed rotation. As a result of this rotation, we get a new path whose endpoint is either v_{i-1} or v_{i+1} (it would be the one which is closer to v' on the path $P_{v_1,v'}$).

Suppose, for example, that the new endpoint is v_{i-1}. The only other possibility for v_{i-1} to be a new endpoint is if we rotate with v_{i-2} as a pivot. Using this fact and Eq.(6.1), the number of new paths on depth $t+1$ is

$$|\,\text{NEW}_{t+1}\,| \geq \left\lceil \frac{|T|}{2} \right\rceil \geq \left\lceil \frac{|N(S_t)| - 2\,|S_t| + 1}{2} \right\rceil , \tag{6.2}$$

and

$$|S_{t+1}| = |S_t| + |\text{NEW}_{t+1}| \geq \left\lceil \frac{|N(S_t)|}{2} \right\rceil \tag{6.3}$$

Moreover, we observe that since P10 holds then when $S_t \leq \frac{n}{\log n}$ then

$$|S_{t+1}| \geq |S_t| + 1 \tag{6.4}$$

In order to analyze the growth rate of S_t, we divide S_t into three sets, LARGE_t, SMALL_t and TINY_t, which are respectively, the set of large vertices, the set of small vertices and the set of tiny vertices in S_t.

As we will analyze precisely, the growth rate of S_t depends on the relative size of the three sets in S_t. For example, if most of the vertices in S_t are *large*, then using property P7, $|S_{t+1}| = \Omega(\log n)|S_t|$. While on the other hand, if most of the vertices in S_t are of constant degree, then the growth rate would be constant.

To do this analysis precisely, in each depth t, we identify a dominant set in S_t, according to the following.

Definition 6.4.2 (Dominant set in S_t) *Let $0 < \xi$ be an arbitrary small constant. Then,*

- *If $|\mathrm{LARGE}_t| \geq \frac{\xi}{2}|S_t|$ then LARGE_t is the dominant set. Otherwise ,*
- *If $|\mathrm{SMALL}_t| \geq \frac{\xi}{2}|S_t|$ then SMALL_t is the dominant set. Otherwise,*
- *If $|\mathrm{TINY}_t| \geq (1-\xi)|S_t|$ then TINY_t is the dominant set.*

Remark 6.4.3 *To make the analysis simpler we will assume that the size of S_t is at least $2\log\log n$. Using Eq.* (6.4) *this is true when the depth of the tree is at most $2\log\log n$.*

Analyzing $|S_{t+1}|$

We next analyze the size of S_{t+1} in each of the cases above. For each case, this is done is three steps. In the first step, we use P7 , P8 and P9 to find the size of the neighborhood of the dominant set. In the second step, we find the size of $N(S_t)$, taking into consideration that some of the neighbors of the dominant set may be in S_t. In the third step, we use Eq. (6.3) to find the size of S_{t+1}.

Size of S_{t+1} when LARGE_t is the dominant set :

We assume that $|\,\mathrm{LARGE}_t\,| \leq \frac{n}{\log n}$.

$$|N(\mathrm{LARGE}_t)| \geq \frac{\log n}{300}|\,\mathrm{LARGE}_t\,| \geq \frac{\log n}{300}\cdot\frac{\xi}{2}|S_t|$$

$$|N(S_t)| \geq |N(\mathrm{LARGE}_t)| - |S_t| \geq \left(\frac{\log n}{300}\cdot\frac{\xi}{2} - 1\right)|S_t| \geq \frac{(\log n)\xi}{700}\cdot|S_t|$$

$$|S_{t+1}| \geq \frac{(\log n)\xi}{1400}\cdot|S_t|$$

Size of S_{t+1} when SMALL_t is the dominant set :

$$|N(\mathrm{SMALL}_t)| \geq (\log\log n)^{c_2-1}\,|\mathrm{SMALL}_t| \geq (\log\log n)^{c_2-1}\tfrac{\xi}{2}\,|S_t|$$

$$|N(S_t)| > (\log\log n)^{c_2}\tfrac{\xi}{2}\,|S_t| - |S_t| = \left((\log\log n)^{c_2-1} - \tfrac{2}{\xi}\right)\tfrac{\xi}{2}\,|S_t|$$

$$|S_{t+1}| \geq \tfrac{1}{2}\left((\log\log n)^{c_2} - \tfrac{2}{\xi}\right)\tfrac{\xi}{2}\,|S_t| \geq (\log\log n)^{c_2-2}\,|S_t|$$

Size of S_{t+1} when TINY_t is the dominant set :

$$|N(\mathrm{TINY}_t)| \geq (2+\epsilon)\,|\mathrm{TINY}_t| \geq (2+\epsilon)(1-\xi)\,|S_t|\ ,$$

for some $0 < \epsilon < 1$.

Remark 6.4.4 *This is true based on P9, Remark 6.4.3 and since we assume that the minimum degree is* 3. *Note that this is the only place in the proof where we rely on the assumption that the minimum degree is* 3 *and not* 2.

Continue with the analysis,

$$|N(S_t)| \geq (2+\epsilon)(1-\xi)\,|S_t| - \xi\,|S_t| = (2+\epsilon-\xi(3+\epsilon))\,|S_t|$$

$$|S_{t+1}| \geq (1+\tfrac{\epsilon}{2}-\xi(1.5+\tfrac{\epsilon}{2}))\,|S_t|$$

By taking $\xi \leq \frac{\epsilon}{10}$,

$$|S_{t+1}| \geq (1+\epsilon/4)\,|S_t|$$

Analyzing the Depth of the Tree

Let $\hat{\tau}$ be the depth of the tree. Let τ' be the number of times that LARGE_t is the dominant set and $\mathrm{LARGE}_t \leq \frac{n}{\log n}$. Then in each such time

$$|S_{t+1}| \geq \frac{(\log n)\xi}{1400}|S_t|.$$

Hence after at most,

$$1 + \left\lceil \frac{\log n - \log\log n}{\log\log n - \log(1400\xi^{-1})} \right\rceil \leq \frac{1.1\log n}{\log\log n}\ ,$$

times that LARGE_t is the dominant set, S_t has at least $\frac{n}{\log n}$ large vertices. Let S_τ be the first time that it happens.

If S_τ does not contain $\frac{n}{\mu}$ vertices, then we pick a set $S \subset S_\tau$ of $\frac{n}{\log n}$ large vertices. Using property P7, $|N(S)| \geq \frac{n}{300}$, and therefore $|N(S_\tau)| \geq \frac{n}{300} - \frac{n}{\mu} \geq 2\frac{n}{\mu}$ (since $\mu = 1000$). Using Eq. (6.3) , $|S_{\tau+1}| \geq \frac{n}{\mu}$.

To complete the analysis of $\hat{\tau}$, we need to count the number of times that SMALL_t and TINY_t are the dominant sets. Let τ'' and τ''' be these numbers.

Analyzing τ'' : Let $t' < t''$ be two levels for which SMALL_t is the dominant set. Then

$$\frac{|\text{SMALL}_{t''}|}{|\text{SMALL}_{t'}|} \geq \frac{\frac{\xi}{2}|S_{t''}|}{|S_{t'}|} = \frac{\xi}{2}(\log\log n)^{c_2-2} \geq (\log\log n)^{c_2-3}$$

Then τ'' satisfies,

$$\left(\frac{\xi}{2}(\log\log n)^{c_2-3}\,|S_t|\right)^{\tau''} \geq n^{1/3}$$

Hence,

$$\tau'' \leq \frac{\log n}{c_2 \log\log\log n}$$

Analyzing τ''' : Let $t' < t''$ be two levels for which TINY_t is the dominant set. Since $\xi \leq \epsilon/10$,

$$\frac{|\text{TINY}_{t''}|}{|\text{TINY}_{t'}|} \geq (1-\xi)(1+\epsilon/4) \geq 1+\epsilon/8$$

Then τ''' satisfies,

$$(1+\epsilon/8)^{\tau'''} \geq e^{\frac{\log n}{\log\log n}}$$

Hence,

$$\tau''' = \frac{1}{\log(1+\epsilon/8)}\frac{\log n}{\log\log n}$$

Hence $\hat{\tau} \leq \tau + 2\log\log n + 1 = \tau' + \tau'' + \tau''' + 2\log\log n + 1$ (the $2\log\log n$ is due to Remark 6.4.3). Since τ'' is the dominating factor,

$$\hat{\tau} \leq 4\tau'' \leq \frac{4\log n}{c_2\log\log\log n} = \frac{\log n}{c_1\log\log\log n} = t_0\ , \tag{6.5}$$

by taking $c_1 = c_2/4$. Note that since c_2 is arbitrary large, so is c_1. This completes the proof of Claim 6.4.1 and of Theorem 4. ■

From the above analysis, we can conclude the following,

Corollary 6.4.5 *Let G be a semi random graph, whose random part is in Γ_1. Let r be the current stage. If during Semi-Ham-Stage$_1$ no simple or cycle extension were done, then $r \geq \frac{n}{\mu}$.*

Proof of Corollary 6.4.5: Recall that the length of the input path to stage r is r. Assume that $r < \frac{n}{\mu}$. Consider the initial path P_r. During the first call to Build-Tree we have proved that there exist t', s.t. $|N(S_{t'})| \geq \frac{n}{\mu}$. Since $r < \frac{n}{\mu}$, then there must be one vertex in $N(S_{t'})$ which is not in P_r, and hence we can perform a simple extension. Contradiction. ■

6.5 Proof of Semi-Ham-Stage$_2$

The idea in Semi-Ham-Stage$_2$ is that among all the paths that have been found in Semi-Ham-Stage$_1$, there is one path, and one intermediate point, for which we succeed to close a cycle. Since we are going over all possible paths, and all possible intermediate points, we would find this path. Then if $r < n$, we can perform a cycle extension in order to find a path of length $r+1$, or if $r = n$, then we have found a Hamiltonian cycle. The main theorem we want to prove is as follows.

Theorem 5 *Let G be a semi random graph whose random part is in Γ_1. Then there exists a pair a, b, where $a \in A$ and $b \in B(a)$, such that for the path $P_{a,b}$, there exists an intermediate point x, for which a cycle is closed during Semi-Ham-Stage$_2$.*

Proof of Theorem 5: We consider an execution of stage r, whose initial path was P_r. Let

$$\rho = 2t_0 = \frac{2\log n}{c_1 \log\log\log n}.$$

We divide P_r into 2ρ segments $I_1, \ldots, I_{2\rho}$ of equal length. Any path $P_{a,b}$ was generated by at most ρ rotations, hence at least half of its segments are unbroken (but may be reversed). Using Corollary 6.4.5, we have that,

Fact 6.5.1 (Number and length of segments)

- *# of segments* $= 2\rho = 4t_0 \le \frac{4\log n}{c_1 \log\log\log n}$
- *length of a segment* $\ge \frac{\frac{n}{\mu}}{\#\ of\ segments} \ge \frac{\frac{n}{\mu}}{\frac{4\log n}{c_1 \log\log\log n}} \ge \frac{c_1 n \log\log\log n}{4\mu \log n}$

Let

$$k = \frac{\log\log n}{6\log\log\log n}\ .$$

We now consider sequences of k segments. Any sequence $\sigma = I_{i_1}, \ldots, I_{i_k}$ of segments specify the order of the k segments in it, and the direction of each segment with respect to P_r, that is any segment can be either forward or reversed.

For every path $P_{a,b}$, any choice of k segments from its unbroken segments would give us a sequence. Since in every path the number of unbroken segments is at least ρ, then every path $P_{a,b}$ contains at least $\binom{\rho}{k}$ sequences.

For a given sequence σ, let $L(\sigma)$ be the set of pairs (a, b), such that for each pair a, b, the path $P_{a,b}$ contains the sequence σ. We get,

$$\sum_{\sigma} L(\sigma) \ge \frac{1}{2}\left(\frac{n}{\mu}\right)^2 \binom{\rho}{k}\ , \tag{6.6}$$

where the $\frac{1}{2}$ is since any path can appear at most twice.

Since any sequence is determined by the k segments it contains, their relative order, and their direction, then the number of different sequences is

$$\binom{2\rho}{k} k! 2^k \ . \tag{6.7}$$

Using Eq. (6.6) and Eq. (6.7), there is a sequence σ_0, for which

$$L(\sigma_0) \geq \frac{\frac{1}{2}\left(\frac{n}{\mu}\right)^2 \binom{\rho}{k}}{\binom{2\rho}{k} k! 2^k} > \frac{1}{2}\left(\frac{n}{\mu}\right)^2 \left(\frac{\rho - k}{2\rho - k}\right)^k \frac{1}{k! 2^k} \tag{6.8}$$

We fix the sequence σ_0. Since $k = \Theta(\frac{\log\log n}{\log\log\log n})$ and $\rho = \Theta(\frac{\log n}{\log\log\log n})$, then for n large enough, $k \leq \rho/2$. Hence $\frac{\rho-k}{2\rho-k} \geq \frac{1}{3}$. Substituting this, we get,

Corollary 6.5.2 *There exists a sequence σ_0 for which,*

$$L(\sigma_0) \geq \frac{1}{2} \cdot \left(\frac{n}{\mu}\right)^2 \frac{1}{k! 6^k} = n^2 \cdot \frac{1}{2\mu^2 k! 6^k} = n^2 \cdot \alpha \ ,$$

where $\alpha = \frac{1}{2\mu^2 k! 6^k}$.

To conclude, up to this point we proved the existence of a sequence σ_0, which appears in αn^2 different paths.

Let

$$\hat{A} = \{a \in A : |\{(a,b) : (a,b) \in L(\sigma_0)\}| \geq \alpha n/2\},$$

and for each $a \in \hat{A}$, let

$$\hat{B}(a) = \{b : (a,b) \in L(\sigma_0)\}.$$

Claim 6.5.3 (Sizes of $\hat{A}$ and $\hat{B}(a)$)

(1) $\left|\hat{A}\right| \geq \frac{\alpha n}{2}$.

(2) $\forall a \in \hat{A}$, $\left|\hat{B}(a)\right| \geq \frac{\alpha n}{2}$.

Moreover,

$$\frac{\alpha n}{2} \geq \frac{20 n \log\log n}{\log n}$$

Proof : Proof is in Appendix B. ■

We now divide the sequence σ_0 into two parts. Let C be the union of the first $k/2$ segments, and let D be the union of the last $k/2$ segments. We note that

$$|C|, |D| \geq \frac{k}{2} \cdot \text{length of a segment}$$

We now define the notion of an interior set.

Definition 6.5.4 (Interior set) *Given a set W of vertices and a path P, we say that $w \in W$ is an interior point of W w.r.t. P, if both neighbors of w along P are in W. The set of interior points is called the interior set of W, and denoted by* $\mathrm{int}(W)$.

We consider the set $\mathrm{int}(C)$ w.r.t. the path P_r. Since C is the union of unbroken segments, then all the points in C which are not endpoints of segments, are in $\mathrm{int}(C)$.

$$\begin{aligned}
|\mathrm{int}(C)| &\geq |C| - 2\frac{k}{2} \\
&\geq \frac{k}{2} \cdot (\text{length of a segment - 2}) \\
&\geq \frac{\log\log n}{12 \log\log\log n} \cdot \left(\frac{c_1 n \log\log\log n}{4\mu \log n} - 2 \right) \\
&\geq \frac{\log\log n}{12 \log\log\log n} \cdot \frac{c_1 n \log\log\log n}{5\mu \log n} \\
&= \frac{c_1 n \log\log n}{60\mu \log n}
\end{aligned}$$

We now claim the following.

Claim 6.5.5 *There is a set $C' \subseteq C$, s.t. $|\mathrm{int}(C')| \geq \frac{1}{2}|\mathrm{int}(C)|$, and for any vertex $c' \in C'$ there are at least $\frac{\log n}{10n}|\mathrm{int}(C)|$ green edges from c' to* $\mathrm{int}(C')$.

Proof of Claim 6.5.5: Recall that subgraph spanned by the green edges, has properties P4, P5 and P11 (see Lemma 6.2.2). We use greedy approach. We start with $C' = C$. Any vertex in C' which have less than $d_1 = \frac{\log n}{10}|\mathrm{int}(C)|$ green edges to $\mathrm{int}(C')$ is deleted from C'. Let R be the set of deleted vertices, $|R| = r$. Any vertex which is deleted from C' decreases the size of $\mathrm{int}(C')$ by at most 3 vertices (since if we take out a vertex from C' then itself and its neighbors along P_r should be removed from $\mathrm{int}(C')$, if they are there). Hence if $r \leq \frac{1}{6}|\mathrm{int}(C)|$ then,

$$|\mathrm{int}(C')| \geq |\mathrm{int}(C)| - 3r \geq \frac{1}{2}|\mathrm{int}(C)| \tag{6.9}$$

and we are done. Assume by contradiction, $r > \frac{1}{6}|\mathrm{int}(C)|$. Consider the sets R and $\mathrm{int}(C')$ after $\frac{1}{6}|\mathrm{int}(C)|$ steps. Both sets are larger than $\frac{100n \log\log n}{\log n}$. Since any vertex in R has less than d_1 green edges to $\mathrm{int}(C')$, we have that

$$e_{\text{green}}(\mathrm{int}(C'), R) \leq |R| \cdot d_1 = |R||\mathrm{int}(C)|\frac{\log n}{10n} \tag{6.10}$$

But according to Lemma 6.2.2 and property P5,

$$e_{\text{green}}(\mathrm{int}(C'), R) \geq |\mathrm{int}(C')|\,|R|\,\frac{\log n}{4n} \geq |R|\,|\mathrm{int}(C)|\,\frac{\log n}{8n}$$

which contradicts Eq. (6.10), and hence completes the proof of Claim 6.5.5. ■

Similarly, we can prove an analogue result for D.

Claim 6.5.6 *There is a set $D' \subseteq D$, s.t. $|\operatorname{int}(D')| \geq \frac{1}{2}|\operatorname{int}(D)|$, and for any vertex $d' \in D'$ there are at least $\frac{\log n}{10n}|\operatorname{int}(D)|$ green edges from d' to $\operatorname{int}(D')$*

By taking c_1 be large enough, the following is concluded from Claim 6.5.5 and Claim 6.5.6,

Corollary 6.5.7

(a) $|\operatorname{int}(C')|, |\operatorname{int}(D')| \geq 17500\frac{n \log\log n}{\log n}$

(b) Any vertex in $\operatorname{int}(C')$ $[\operatorname{int}(D')]$ *has at least* $3500 \log\log n$ *green edges to* $\operatorname{int}(C')$ $[\operatorname{int}(D')]$.

Which leads to the following claim,

Claim 6.5.8

- *There is a vertex $\hat{a} \in \hat{A}$, which has an edge to* $\operatorname{int}(C')$
- *There is a vertex $\hat{b} \in \hat{B}(\hat{a})$, which has an edge to* $\operatorname{int}(D')$.

Proof of Claim 6.5.8: The claim follows from Corollary 6.5.7, Claim 6.5.3, and property P5. ■

We now reach the last step of the proof. Let $\hat{a}$,$\hat{b}$ be the two vertices guaranteed from Claim 6.5.8 and consider the path $P_{\hat{a},\hat{b}}$. Pick some vertex x on $P_{\hat{a},\hat{b}}$, which separates C' and D' along $P_{\hat{a},\hat{b}}$. This divides the path into two sub-paths $P_L = (\hat{a}, \ldots, x)$ and $P_R = (x, \ldots, \hat{b})$. Recall that during the execution of Semi-Ham-Stage$_2$, we call for the procedure Build-Tree-2 twice. One time with P_L, and second time with P_R. In both calls x is the fixed endpoint. As a result of the calls, we generate two sets $L \subseteq V(P_L)$ and $R \subseteq V(P_R)$ of other endpoints. If these sets would be large enough, then there must be an edge (l, r), with $l \in L$, and $r \in R$. Then we can combine the paths $P_{l,x}$, $P_{x,r}$ and the edge (l, r) into a cycle. We claim that these sets are indeed large enough.

Claim 6.5.9 (Size of L and R)

1. $|L| \geq \frac{3n \log\log n}{\log n}$
2. $|R| \geq \frac{3n \log\log n}{\log n}$

Proof of Claim 6.5.9: We prove for L, same proof holds for R. Define

$$U_i = \left\{ v : \begin{array}{l} v \text{ is the endpoint of path obtained by } \textbf{at most } i \text{ rotations,} \\ \text{with fixed endpoint } x \text{ and all broken edges in } P_L \end{array} \right\}$$

We consider only the vertices in C' in our analysis. Let $U'_i = U_i \cap C'$. We would prove that,

(1) $U'_1 \neq \emptyset$

(2) As long as $U'_i \leq \frac{3\log\log n}{\log n}$, $\left|U'_{i+1}\right| \geq 3\left|U'_i\right|$.

Since $U'_i \subseteq L$, and the depth of the tree in the procedure Build-Tree-2 is bounded by $\log n$, these two assertions would prove Claim 6.5.9.

Using Claim 6.5.8, there is an edge $(\hat{a}, c)$ where $c \in \mathrm{int}(C')$. Let c' be the neighbor of c along P_L which is closer to $\hat{a}$. Then c' is the new endpoint, after rotating with $(\hat{a}, c)$ as the broken edge. Moreover, $c' \in C'$, and (1) is proven.

Define $W_i = N(U_i) \cap \mathrm{int}(C')$. Using Corollary 6.5.7, any vertex $u \in U'_i$ has at least $3500\log\log n$ green edges to W_i, hence

$$e_{\text{green}}(U'_i, W_i) \geq |U'_i| \cdot 3500\log\log n \tag{6.11}$$

Assume both U'_i and W_i are smaller than $\frac{30n\log\log n}{\log n}$. if $|W_i| \leq 10|U'_i|$ then according to Lemma 6.2.2 and P11,

$$e_{\text{green}}(U'_i, W_i) \leq 300(\log\log n)\max\{|W_i|, |U_i|\} \leq 3000\log\log n\,|U_i|$$

which contradicts Eq. (6.11). Therefore,

$$|W_i| \geq 10|U'_i| \tag{6.12}$$

For a vertex $w \in W_i$, let w_r and w_l be respectively, its right neighbor and its left neighbor along P_L. Define,

$$T'_i = \{w \in W_i \ : w, w_l \text{ and } w_r \notin U_i\}$$

Since any vertex in $U_i \backslash C'$ do not have a neighbor along P_L which is $\mathrm{int}(C')$, then

$$|T'_i| \geq |W_i| - 3\,|U'_i| \geq 7\,|U'_i| \ .$$

Since $w_l, w_r \in C'$, then using similar arguments as in the proof of Eq. (6.2) and Eq. (6.3),

$$\left|U'_{i+1}\right| \geq |U'_i| + \left\lceil \frac{|T'_i|}{2} \right\rceil \geq 3\,|U'_i|$$

To complete the proof, we consider the case that either U'_i or W_i are larger than $\frac{30n\log\log n}{\log n}$. If it is U'_i then we are done. Otherwise, if it is W_i, then $T'_i \geq \frac{27n\log\log n}{\log n}$, and in the next stage $|U'_{i'+1}| \geq \frac{12n\log\log n}{\log n}$ (it also holds that $\left|U'_{i+1}\right| \geq 3|U'_i|$). Hence assertion (2) is proven, which completes the proof of Claim 6.5.9. ■

Theorem 5 is implied from Claim 6.5.9 and property P4. ■

6.6 Expected running time of Semi-Ham-Stage$_2$

In Section 5.5.1, a randomized version of Semi-Ham-Stage$_2$ was suggested, in order to improve the expected running time. The justify the analysis there, we analyze the number of triples (a, b, x), where $a \in A, b \in B(a), x \in P_{a,b}$, for which Semi-Hamsucceeds to close a cycle.

Claim 6.6.1 *There are* $\Omega\left(\frac{n^3(\log\log n)^5}{(\log n)^3}\right)$ *triples for which Semi-Ham succeeds to close a cycle.*

Proof of Claim 6.6.1: We start by extending Claim 6.5.8.

Claim 6.6.2

- $\left|\left\{\hat{a} : \hat{a} \in \hat{A} \textit{ and } \hat{a} \textit{ has an edge to } \mathrm{int}(C')\right\}\right| \geq \Omega\left(\frac{n(\log\log n)^2}{\log n}\right)$
- *for each* $\hat{a}$, $\left|\left\{\hat{b} : \hat{b} \in \hat{B}(\hat{a}) \textit{ and } \hat{b} \textit{ has an edge to } \mathrm{int}(D')\right\}\right| \geq \Omega\left(\frac{n(\log\log n)^2}{\log n}\right)$

That is, instead of claiming that there exists one such pair, we claim that the number of pairs is $\Omega\left(\frac{n^2(\log\log n)^4}{(\log n)^2}\right)$.

Proof of Claim 6.6.2: The claim follows from Corollary 6.5.7, Claim 6.5.3 and property P5. ■

Recall that the sets C and D were defined by dividing the k segments of σ_0 into two. Instead, we can define C as the union of the first $k/3$ segments in σ_0, and define D as the union of the last $k/3$ segments in σ_0. Then by slightly changing the constants, the rest of the proof remains valid. This change would leave us with $\Omega\left(\frac{n\log\log n}{\log n}\right)$ valid intermediate points for each path $P_{\hat{a},\hat{b}}$. Combining this with Claim 6.6.2 completes the proof of Claim 6.6.1. ■

Chapter 7

Handling Vertices of Degree 2

In this chapter we present the changes needed to be done in the algorithm and in the proof, in order to adapt it to the case that the random part of the semi random graph has vertices of degree 2.

We first present an example which demonstrates the problem, then we present the solution, and then prove its correctness.

7.1 The Problem

In Theorem 3 it was claimed that if a semi-random graph has its random part in Γ_1, then Semi-Ham will find a Hamiltonian cycle in it. We give an example to show that in the case that the random graph has vertices of degree 2, this is not necessarily true. We use similar example to Figure 4.1. Details follow.

Let G be a graph with a vertex set $[n]$. Assume m is close to the threshold, and hence (see Appendix A) the number of vertices of degree 2 is $\Theta(\log n)$. Let t be this number.

Assume that set of vertices of degree 2 is $\text{TWO} = \{5i : 1 \le i \le t\}$. For the vertex $5t$, let its neighbors be $5t-1$, and $5t-6$. For each other vertex v of degree 2, $v = 5i$, let its neighbors be $5i-6$ and $5i+1$. Assume there is a fake edge between any two vertices j and $j+1$, for $j < 5t$.

During the execution of Semi-Ham, in every stage till stage $5t$, there will be an extension in the beginning of the stage. In the beginning of stage $5t$, the initial path P will be as in Figure 7.1. Then, in the call for Build-Tree, with

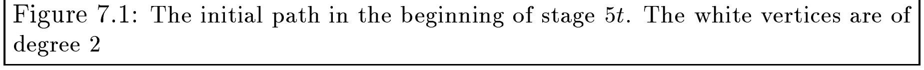

Figure 7.1: The initial path in the beginning of stage $5t$. The white vertices are of degree 2

Fixed-Semi-Ham-Stage$_1$

Input : A path P_{v_1,v_r} of length r.

1. Call Semi-Ham-Stage$_1$, with the path P_{v_1,v_r}, with no depth limit on Build-Tree.
2. Let $\mathcal{P}$ be the set of paths returned by the call.
3. For each path $P \in \mathcal{P}$,
   a. Call Semi-Ham-Stage$_1$, with the path P as an initial path (this time, with the depth limit).
   b. If Semi-Ham-Stage$_1$ succeeded, then terminate and move to Semi-Ham-Stage$_2$.
4. If all the calls to Semi-Ham-Stage$_1$ failed, then fail.

Table 7.1: Fixed-Semi-Ham-Stage$_1$, a modified version of Semi-Ham-Stage$_1$, to handle vertices of degree 2

P as the path, and 1 as the fixed endpoint, the depth of the tree would be $\Theta(\log n)$.

7.2 The Solution

The solution to the problem is achieved by modifying the first part, Semi-Ham-Stage$_1$ into Fixed-Semi-Ham-Stage$_1$. The second part, Semi-Ham-Stage$_2$ remains unchanged.

The key idea is to find a path for which the original Semi-Ham-Stage$_1$ succeeds even if there are vertices of degree 2. In order to find this path, we first call the original Semi-Ham-Stage$_1$ with no depth limit. This gives us a set $\mathcal{P}$ of $\Theta(n^2)$ paths. Then for each path $P \in \mathcal{P}$, we call Semi-Ham-Stage$_1$ again (with the original depth limit). We will prove that for at least one path we will not fail. The pseudo code of Fixed-Semi-Ham-Stage$_1$ is given in Table 7.1.

We remark that this version of the algorithm should be used even if none of the vertices in the input graph has degree 2, since even in this case, the random part may have vertices of degree 2.

7.3 Proof of Fixed-Semi-Ham-Stage$_1$

The main assertion we want to prove is as follows.

Theorem 6 *Let G be a semi-random graph whose random part is in Γ_1. Let $\mathcal{P}$ be as defined above. Then, there exists a path $P \in \mathcal{P}$ for which Semi-Ham-Stage$_1$*

will not fail.

Observe that the assumption about G ensures that if Semi-Ham-Stage$_1$ succeeds, then Semi-Ham-Stage$_2$ will also succeed.

Proof of Theorem 6: Throughout the proof we assume that the input graph G is such that its random part is in Γ_1.

Recall that during the execution of Semi-Ham-Stage$_1$, the procedure Build-Tree is called $O(n)$ times, and all of these calls should not fail. We start by proving some sufficient conditions for the success of Build-Tree.

Claim 7.3.1 *Let P be an initial path to Build-Tree. If during the execution of Build-Tree, the size of S_2 is at least $10\,|\mathrm{TWO}|$, then the procedure will succeed.*

Proof of Claim 7.3.1: Recall that the only place in the proof in Chapter 6 where we used the fact that the minimum degree is 3 is when we analyzed $N(\mathrm{TINY}_t)$ when TINY_t was the dominant set (see Remark 6.4.4). If TINY_t is the dominant set, then by taking $\xi < \frac{1}{4}$,

$$|\mathrm{TINY}_t| \geq (1-\xi)\,|S_t| \geq (1-\xi)10\,|\mathrm{TWO}| \geq 5\,|\mathrm{TWO}| \ .$$

Hence, at most $\frac{1}{4}$ of the vertices in TINY are small and we can apply P9 to get $|N(\mathrm{TINY})| \geq 2.25\,|\mathrm{TINY}|$. Since the rest of the analysis in the proof remains unchanged, this proves Claim 7.3.1. ■

We turn now to analyze the call to Semi-Ham-Stage$_1$ with no depth limit (see Table 7.1, line 1). During this call, the procedure Build-Tree is called $O(n)$ times, each time with no depth limit. We claim that each of these calls succeeds.

Claim 7.3.2 *During every call to Build-Tree with no depth limit, the size of the set of other endpoints that were found is at least $\frac{n}{\mu}$.*

Proof of Claim 7.3.2: In Section 6.4, Eq. (6.4) it was proven that if $|S_t| \leq \frac{\log n}{n}$ then $|S_{t+1}| \geq |S_t| + 1$.

Hence when TINY_t is the dominant set, the size S_{t+1} will be at least $|S_t|+1$. Since the depth of the tree is unbounded and the rest of the analysis in Section 6.4 remains unchanged (as explained in the proof of Claim 7.3.1) this proves Claim 7.3.2. ■

Claim 7.3.2 implies the following corollary.

Corollary 7.3.3 *In the end of the call of Semi-Ham-Stage$_1$ with no depth limit, we have,*

- *A set A of size $\frac{n}{\mu}$.*
- *For each $a \in A$, a set $B(a)$ of size $\frac{n}{\mu}$.*
- *For each $a \in A$, $b \in B(a)$ there is a path $P_{a,b}$ in $\mathcal{P}$.*

These paths are the candidates for Semi-Ham-Stage$_1$. To finish the proof, we need to prove that at least one call to Semi-Ham-Stage$_1$ (in line 3a, Table 7.1) will succeed on at least one of these paths.

Claim 7.3.4

(1) There is a vertex $a' \in A$, such that for each path P whose endpoints are a' and v (for any vertex v) when Build-Tree is called with P and with v as the fixed endpoint, $|S_2| \geq 10\,|\mathrm{TWO}|$.

(2) With high probability the number of a' satisfying (1) is $\Omega(n)$.

Proof of Claim 7.3.4: Define the following sets,

$$\mathrm{BAD} = \{v : v \notin \mathrm{LARGE} \wedge \deg_{\mathrm{semi}}(v) \leq n^{1/3}\},$$

$$\overline{\mathrm{BAD}} = V \backslash \mathrm{BAD} = \{v : v \in \mathrm{LARGE} \vee \deg_{\mathrm{semi}}(v) > n^{1/3}\},$$

Since $|\mathrm{BAD}| \leq n^{1/3}$ and since the degree of each vertex in BAD is bounded then $|N^*(\mathrm{BAD})| \leq 2n^{2/3}$.

We note that any vertex which is not in $N^*(\mathrm{BAD})$ must have all its neighbors in $\overline{\mathrm{BAD}}$. Hence all its neighbors are either large vertices or their degree in the semi-random graph is at least $n^{1/3}$.

Let a' be a vertex in A, assume $|N(a') \cap \overline{N^*(\mathrm{BAD})}| \geq \frac{\log n}{400}$. That is, at least $\frac{\log n}{400}$ of the neighbors of a' are not in $N^*(\mathrm{BAD})$.

Consider an execution of Build-Tree on $P_{a',v}$ with v as the fixed endpoint (v is any vertex). Since a' has at least $\frac{\log n}{400}$ neighbors which are not in $N^*(\mathrm{BAD})$, then in S_1 at least $\frac{\log n}{400} - 1$ of the endpoints are in $\overline{\mathrm{BAD}}$. If any of these endpoints has degree at least $n^{1/3}$ then P3 implies $|S_2| \geq 10\,|\mathrm{TWO}|$ and we are done. Otherwise, all these endpoints are large, hence

$$|N(S_1)| \geq \frac{\log n}{300}\left(\frac{\log n}{400} - 1\right) - |S_1| \geq \frac{(\log n)^2}{15000}\ ,$$

for n large enough. Using Eq. (6.3) and property P3,

$$|S_2| \geq \frac{(\log n)^2}{30000} \geq 10\,|\mathrm{TWO}|\ .$$

To complete the proof of (1) we need to prove the existence of such a'. Let $A_1 \subset A$ be a set of large vertices of size $\frac{n}{\log n}$. Then $|N(A_1)| \geq \frac{n}{300}$, and $|N(A_1) \cap \overline{N^*(\mathrm{BAD})}| \geq \frac{n}{400}$. Using averaging at least one vertex A_1 has $\frac{\frac{n}{400}}{\frac{n}{\log n}} = \frac{\log n}{400}$ neighbors in $\overline{N^*(\mathrm{BAD})}$, which proves (1).

To prove (2) let A' be the set of a'-s satisfying (1). Consider the edge probability of the random graph. If $p \geq \frac{(\log n)^2}{n}$ then Claim A.4.1 implies that whp all vertices in G are large, and therefore $N^*(\mathrm{BAD}) = \emptyset$ and $A' = A$.

If $p \leq \frac{(\log n)^2}{n}$, define the set A'' as the set of large vertices which are not in A'. Then each vertex in A'' has at least $\frac{\log n}{20} - \frac{\log n}{400} > \frac{\log n}{40}$ neighbors in $N^*(\mathrm{BAD})$. If $|A''| \geq 2n^{2/3}$ fix a subset $A_1'' \subseteq A''$ of size $2n^{2/3}$. Then

$$e(A_1'' \cup N^*(\mathrm{BAD})) \geq e(A_1'', N^*(\mathrm{BAD})) \geq \frac{\log n}{40}\,|A_1''| \geq 100\,|A_1'' \cup N^*(\mathrm{BAD})|$$

which according to Claim A.4.7 occurs with probability $o(1)$. Then any large vertex in $A \backslash A''$ is in A', and therefore $|A'| = \Omega(n)$. ■

Similar arguments can be used to prove an analogous result for $B(a')$.

Claim 7.3.5 *Let a' be a vertex satisfying Claim 7.3.4. Then,*

(1) There is a vertex $b' \in B(a')$, such that for each path P whose endpoints are b' and v (for any vertex v), when Build-Tree is called with P and with v as the fixed endpoint, $|S_2| \geq 10\,|\mathrm{TWO}|$.

(2) With high probability the number of b' satisfying (1) is $\Omega(n)$.

To complete the proof, consider any path $P_{a',b'}$. Consider the call to Semi-Ham-Stage$_1$ with $P_{a',b'}$ as the initial path (see line 3a, Table 7.1).

During Semi-Ham-Stage$_1$, the first call to Build-Tree is with $P_{a',b'}$ as the initial path and with a' as the fixed endpoint. Then relying on Claim 7.3.5 and on Claim 7.3.1, this call succeeds.

Observe that Claim 7.3.5 is much stronger than needed. It claims about any path whose endpoint (which is not the fixed one) is b', but it was enough to claim only on the path $P_{a',b'}$.

As a result of the call to Build-Tree, we obtain a set of $\frac{n}{\mu}$ paths, all have a' as an endpoint. For each of these paths, we call to Build-Tree with the endpoint which is not a' as the fixed endpoint. Then Claim 7.3.4 and Claim 7.3.1 imply that all these calls succeed, and hence Semi-Ham-Stage$_1$ succeed. This completes the proof of Theorem 6. ■

7.4 Time Complexity of Fixed Semi-Ham

As was done in Section 5.5.1, we count the number of trials. In Fixed-Semi-Ham-Stage$_1$ we try $O(n^2)$ initial paths, hence the complexity of Fixed-Semi-Ham-Stage$_1$ is $O(n^5)$. The complexity of Semi-Ham-Stage$_2$ remains $O(n^5)$, and hence the complexity of Semi-Ham remains $O(n^6)$.

7.4.1 Randomized Approach

As in Section 5.5.1, a randomized approach can be used. That is, instead of iterating over the paths in $\mathcal{P}$ in an arbitrary order, we iterate in random order. Claim 7.3.4 and 7.3.5 imply that whp for a constant fraction of the paths in $\mathcal{P}$, Semi-Ham will succeed. This implies that the expected number of paths we try until we succeed is a constant. Hence the expected running time of Fixed-Semi-Ham-Stage$_1$ is $O(n^3)$ and the expected running time of Semi-Ham-Stage$_2$ is $O(\mathrm{polylog}(n) \cdot n^2)$ (see section 5.5.1). Hence the expected running time of Semi-Ham remains $O(n^4)$.

Chapter 8

Summary

In this work we have presented the algorithm Semi-Ham, a polynomial time algorithm for finding a Hamiltonian cycle in semi-random graphs.

The motivation for presenting this algorithm was to get over the potential problems that may occur when running the algorithm HAM on semi-random inputs.

To analyze the performance of Semi-Ham we gave a list of graph properties and proved that for a random graph $G_{n,m}$ with $m = \frac{n}{2}(\log n + \log\log n + c_n)$, if $c_n \nrightarrow \infty$ then $G_{n,m}$ a.s. has these properties. Then we have proved that for any graph G, if G has these properties and has minimum degree 2 then Semi-Ham will find a Hamiltonian cycle in any graph generated from G by adding edges.

In fact, we have proved that if $m = \frac{n}{2}(\log n + \frac{1}{2}\log\log n)$, then the list of properties is valid for a.e. graph in $\mathcal{G}_{n,m}$. Since, if $m = \frac{n}{2}(\log n + \frac{1}{2}\log\log n)$ then $G_{n,m}$ a.s. has $\delta(G_{n,m}) < 2$, then we can state our result in terms of graph process, as follows.

Theorem 7 *Let $\tilde{G} = G_0, G_1, \ldots, G_N$ be a graph process. Let $\tau = \tau(\tilde{G}, \text{'}\delta \geq 2\text{'})$. Let G be the graph generated by adding arbitrary edges to G_τ. Then Semi-Ham a.s. finds a Hamiltonian cycle in G, where probability is taken over the choices of graph processes.*

Some other questions that can be explored are as follows. One question is about directed graphs. In this case, the semi-random graph is generated by taking a random directed graph $D_{n,m}$, which is a.s. Hamiltonian, and adding to it arbitrary edges. Then the question is whether we can devise an algorithm that finds a Hamiltonian cycle in such a graph. This question is expected to be more difficult because of the nature of directed graphs, and since the subject of Hamiltonian cycles in random directed graphs was less explored.

A different type of question could arise when using different semi random models. For example, we can consider a model of planted solution. That is, we start with a Hamiltonian cycle, then add random edges, then the adversary may delete edges as long as they are not part of the Hamiltonian cycle. Then, our task is to find a Hamiltonian cycle, not necessarily the original one.

Acknowledgments

I wish to express my deep gratitude to my advisor Prof. Uriel Feige for his guidance and help.
I thank Prof. Benny Sudakov for his help in the preliminary stages of the work.

Bibliography

[1] M. Ajtai, J. Komlós, and E. Szemerédi. First occurrence of Hamilton cycles in random graphs. In B.R. Alspach and C.F. Godsil, editors, *Cycles in graphs, Annals of Discrete Mathematics 27*, pages 173–178. North-Holland, 1985.

[2] A. Blum and J. Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19:204–234, 1995.

[3] Tom Bohman, Alan Frieze, and Ryan Martin. How many random edges make a dense graph Hamiltonian? *Pre-print version.*

[4] B. Bollobas. The evolution of sparse graphs. *Graph Theory and Combinatorics (Proc. Cambridge Combinatorics Conference in Honors of Paul Erdôs) Academic Press*, pages 35–57, 1984.

[5] B. Bollobas, T.I. Fenner, and A.M. Frieze. An algorithm for finding Hamilton paths and cycles in random graphs. *Combinatorica*, 7(4):327–341, 1987.

[6] A.Z. Broder, A.M. Frieze, and E. Shamir. Finding hidden Hamilton cycles. *Random Structures and Algorithms*, 5:395–410, 1994.

[7] Thomas H. Cormen, Charles E. Leiserson, and Ronald E. Rivest. *Introduction to Algorithms.* MIT Press, 1990.

[8] P. Erdôs and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci.*, 5:17–61, 1960.

[9] P. Erdôs and A. Rényi. On the strength of connectedness of a random graph. *Acta. Math. Acad. Sci. Hungar.*, 12:261–267, 1963.

[10] U. Feige and J. Kilian. Heuristics for semirandom graph problems. *Journal of Computer and System Sciences*, To appear.

[11] Alan Frieze and Colin Cooper. Multicoloured Hamilton cycles in random graphs ; an anti-ramsey threshold. *The electronic journal of combinatorics*, 19(4), 1995.

[12] A.M. Frieze. An algorithm for finding Hamilton cycles in random digraphs. *Journal of Algorithms*, 9:181–204, 1988.

[13] A.M. Frieze and C. McDiarmid. Algorithmic theory of random graphs. *Random Structures and Algorithms*, 10, 1997.

[14] A.M. Frieze and J. Yukich. Probabilistic analysis of the traveling salesman problem. *To appear.*

[15] S. Janson, T. Luczak, and A. Rucinski. *Random Graphs.* Wiley Interscience, 2000.

[16] J. Komlós and E. Szemerédi. Limit distributions for the existence of Hamilton circuits in random graphs. *Discrete Mathematics*, 43:55–63, 1983.

[17] M. Krivelevich and B. Sudakov. Sparse pseudo-random graphs are Hamiltonian. To appear.

[18] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. *The Traveling Salesman Problem.* John Wiley & Sons, Chichester, 1985.

[19] L. Posa. Hamiltonian circuits in random graphs. *Discrete Mathematics*, 14:359–364, 1976.

[20] A. Thomason. A simple linear expected time algorithm for finding a Hamilton path. *Discrete Mathematics*, 75:373–379, 1989.

Appendix A

Proof of properties P1-P11

A.1 Introduction

In order to prove the properties we work in the $\mathcal{G}_{n,p}$ model, with $p = m/N$ ($N = \binom{n}{2}$ is the number of edges in the complete graph with n vertices). Claim 2.1.1 then implies the correctness in the $\mathcal{G}_{n,m}$ model. For increasing properties, we prove the correctness for p', then it would imply that it is true for $p \geq p'$. Same holds for $p \leq p'$ for decreasing properties.

We shall use the following bounds during the proof.

- (Binomial distribution) Let $B(n,p)$ be a Binomial random variable with parameters n and p. Then

$$\mathbb{P}[|B(n,p) - np| \geq \epsilon np] \leq 2e^{\frac{-\epsilon^2 np}{3}} \qquad 0 \leq \epsilon \leq 1 \tag{A.1}$$

$$\mathbb{P}[B(n,p) \geq \alpha np] \leq \left(\frac{e}{\alpha}\right)^{\alpha np} \cdot e^{-np} \qquad 1 < \alpha \tag{A.2}$$

- $$\binom{n}{k} \leq \frac{n^k}{k!} \leq \left(\frac{en}{k}\right)^k \tag{A.3}$$

- $$1 - p \leq e^{-p} \quad \text{for } 0 \leq p \leq 1 \tag{A.4}$$

The proof of (A.1) and (A.2) can be found e.g. in [15, Chapter 2]. (A.3) is proved using Stirling approximation (See e.g. [7, Chapter 6]). (A.4) is proved using Taylor expansion.

We assume that n is large enough such that all the approximations in the proofs are valid. Property P6 was proven by Erdôs and Rényi in [9]. The proofs are based on [5, Lemma 3.1] and on [11].

For an event $\mathcal{E}$, we write $\mathbb{P}[\mathcal{E}] = o\left(\frac{1}{\text{poly}}\right)$, if for any polynom p, $\mathbb{P}[\mathcal{E}] = o\left(\frac{1}{p(n)}\right)$.

A.2 Degree Properties

A.2.1 Proof of Property P1

Claim A.2.1 *Let $G_{n,p}$ be a random graph with $p \geq \frac{\log n}{n}$, then*

$$\mathbb{P}[G_{n,p} \notin P1] = o\left(\frac{1}{poly}\right)$$

Proof of Claim A.2.1: We prove for $p = \frac{\log n}{n}$. If $P1$ does not hold, then there is a set S of vertices, $|S| = s = n^{1/3}$, such that every vertex S has at most $\frac{\log n}{20}$ neighbors in V. In order to make these events independent for the vertices in S, we use $V \backslash S$ instead of V. Hence

$$\begin{aligned}
\mathbb{P}[G_{n,p} \notin P1] &\leq \binom{n}{s}\left[\sum_{k=0}^{(\log n)/20}\binom{n-s}{k}p^k(1-p)^{n-s-k}\right]^s \\
&\leq \left[\left(\frac{ne}{s}\right)\frac{\log n}{20}\left(\frac{20e(n-s)}{\log n}\right)^{\frac{\log n}{20}}\left(\frac{\log n}{n}\right)^{\frac{\log n}{20}} e^{-\frac{\log n}{n}\cdot(n-s-\frac{\log n}{20})}\right]^s \\
&\leq \left[(\log n)\cdot n^{\frac{2}{3}}\cdot n^{\frac{\log(20e)}{20}}\cdot e^{-\log n}(1+o(1))\right]^s \\
&\leq \left[(\log n)\cdot n^{\frac{2}{3}+\frac{1}{5}-1}\right]^s \leq n^{-0.1s} = o\left(\frac{1}{\text{poly}}\right)
\end{aligned}$$

■

A.2.2 Proof of Property P2

Claim A.2.2 *Let $G_{n,p}$ be a random graph with $p \geq \frac{\log n}{n}$, then*

$$\mathbb{P}[G_{n,p} \notin P2] = o\left(\frac{1}{poly}\right)$$

Proof of Claim A.2.2: Let $p = \frac{\log n}{n}$. We Use similar arguments as in Claim A.2.1. If P2 does not hold then there is a set of vertices S, $|S| = s = n^{1/\log\log n}$, such that each vertex in S has at most $(\log\log n)^{c_2}$ neighbors in $V \backslash S$. Hence,

$$\begin{aligned}
\mathbb{P}[G_{n,p} \notin P2] &\leq \binom{n}{s} \left[\sum_{k=0}^{(\log\log n)^{c_2}} \binom{n-s}{k} p^k (1-p)^{n-s-k} \right]^s \\
&\leq \left[\frac{ne}{s} \cdot (\log\log n)^{c_2} \left(\frac{en}{(\log\log n)^{c_2}} \cdot \frac{\log n}{n} \right)^{(\log\log n)^{c_2}} e^{-\frac{\log n}{n}(n-s-(\log\log n)^{c_2})} \right]^s \\
&\leq \left[\frac{ne}{s} \cdot (\log\log n)^{c_2} \cdot (\log n)^{(\log\log n)^{c_2}} e^{-\log n} (1+o(1)) \right]^s \\
&\leq \left[\frac{1}{s} \cdot (\log n)^{(\log\log n)^{c_2+1}} \right]^s = \left[n^{-\frac{1}{\log\log n}} \cdot n^{\frac{(\log\log n)^{c_2+2}}{\log n}} \right]^s \\
&< n^{\frac{-s}{2\log\log n}} = o\left(\frac{1}{\text{poly}}\right)
\end{aligned}$$

■

A.2.3 Proof of property P3

Claim A.2.3 *Let $G_{n,p}$ be a random graph with $p \geq \frac{\log n + \epsilon \log\log n}{n}$, where $0 < \epsilon$ is arbitrary small constant. Then*

$$\mathbb{P}[G_{n,p} \notin P3] = o\left(\frac{1}{poly}\right)$$

Proof of Claim A.2.3: Let $\alpha = 2 - \epsilon/2$. We prove that the size of TWO is at most $(\log n)^\alpha$, since $\alpha < 2$, this will prove the claim.

We use similar arguments as in Claim A.2.1. If $|\,\text{TWO}\,| \geq (\log n)^\alpha$ then there exists a set S, $|S| = s = (\log n)^\alpha$, such that each vertex in S has at most 2 vertices in $V \backslash S$. Then,

$$\begin{aligned}
\mathbb{P}[G_{n,p} \notin P3] &\leq \binom{n}{(\log n)^\alpha} \left[\sum_{i=0}^{2} \binom{n-(\log n)^\alpha}{i} p^i (1-p)^{n-s-i} \right]^{(\log n)^\alpha} \\
&\leq \left[\frac{ne}{(\log n)^\alpha} \cdot 3 \cdot \frac{n^2}{2} \cdot \frac{(\log n + \epsilon \log\log n)^2}{n^2} \cdot e^{-\log n - \epsilon \log\log n} (1+o(1)) \right]^{(\log n)^\alpha} \\
&\leq \left[40 \cdot \frac{(\log n)^2}{(\log n)^\alpha \cdot (\log n)^\epsilon} \right]^{(\log n)^\alpha} \leq \left[(\log n)^{-\epsilon/3} \right]^{(\log n)^\alpha} = o\left(\frac{1}{\text{poly}}\right)
\end{aligned}$$

■

We next give a lower bound on $\mathbb{E}|\text{TWO}|$ when $p = \frac{\log n + \log\log n}{n}$. Using that when $p \to 0$ then $(1-p)^n \geq \frac{1}{2} e^{-np}$, then,

$$\begin{aligned}
\mathbb{E}(|\text{TWO}|) &\geq n \cdot \binom{n-1}{2} \left(\frac{\log n + \log\log n}{n} \right)^2 \left(1 - \frac{\log n + \log\log n}{n} \right)^{n-3} = \\
&\geq n \cdot \frac{n^2}{4} \frac{(\log n)^2}{n^2} \cdot \tfrac{1}{2} e^{-\log n - \log\log n} > \frac{\log n}{10}
\end{aligned}$$

A.3 Increasing Properties

A.3.1 Proof of Property P4

Claim A.3.1 *Let $G_{n,p}$ be a random graph with $p \geq \frac{\log n}{n}$. Then,*

$$\mathbb{P}[G_{n,p} \notin P4] = o\left(\frac{1}{poly}\right)$$

Proof of Claim A.3.2: We prove for $p = \frac{\log n}{n}$. The probability that two sets of size $s = \frac{3n \log\log n}{\log n}$ exist with no edge connecting them is at most :

$$\binom{n}{s}\binom{n}{s} \cdot (1-p)^{s^2} \leq \left(\frac{ne}{s}\right)^s \left(\frac{ne}{s}\right)^s e^{-ps^2} \leq \left(\frac{ne}{s} e^{-ps/2}\right)^{2s}$$

Using $\frac{n}{s} = \frac{\log n}{3 \log\log n}$ and $ps = 3 \log\log n$, we get,

$$\mathbb{P}[G_{n,p} \notin P4] \leq \left(\frac{e \log n}{3 \log\log n} e^{-1.5 \log\log n}\right)^{2s} \leq (\log n)^{-s} = o\left(\frac{1}{\text{poly}}\right)$$

■

A.3.2 Proof of property P5

Claim A.3.2 *Let $G_{n,p}$ be a random graph with $p \geq \frac{\log n}{n}$. Then,*

$$\mathbb{P}[G_{n,p} \notin P5] = o\left(\frac{1}{poly}\right)$$

Proof of Claim A.3.2: We prove for $p = \frac{\log n}{n}$. We consider two disjoint sets A and B, with $a = |A|, b = |B| \geq \frac{20n \log\log n}{\log n}$. The expected number of edges between A and B is

$$\mathbb{E}(e(A,B)) = abp = ab\frac{\log n}{n}$$

For fixed A and B, using Chernoff bound (Eq. (A.1)), we get

$$\mathbb{P}\left[e(A,B) \leq ab\frac{\log n}{4n}\right] = \mathbb{P}\left[e(A,B) \leq \frac{1}{4}\mathbb{E}(e(A,B))\right] \leq 2e^{-\frac{9}{48}abp}$$

Using $\frac{n}{a}, \frac{n}{b} \leq \frac{\log n}{20 \log\log n}$ and $bp \geq 20 \log\log n$, we get,

$$
\begin{aligned}
\mathbb{P}[G_{n,p} \notin P5] &\leq \sum_b \sum_{a>b} \binom{n}{a}\binom{n}{b} 2e^{-\frac{9}{48}abp} \\
&\leq 2\sum_b \sum_{a>b} \left(\frac{en}{a}\right)^a \left(\frac{en}{b}\right)^b e^{-\frac{9}{48}abp} \\
&\leq 2\sum_b \sum_{a>b} \left(\frac{e\log n}{20\log\log n}\right)^a \left(\frac{e\log n}{20\log\log n}\right)^b e^{-\frac{9}{48}abp} \\
&\leq 2\sum_b \sum_{a>b} \left(\left(\frac{e\log n}{20\log\log n}\right)^2 e^{-\frac{9}{48}bp}\right)^a \\
&\leq 2\sum_b \sum_{a>b} \left(\frac{\log n}{\log\log n} e^{-\frac{180}{64}\log\log n}\right)^{2a} \\
&\leq 2n^2 \left(\frac{1}{\log n \log\log n}\right)^{\frac{40n\log\log n}{\log n}} = o\left(\frac{1}{\text{poly}}\right)
\end{aligned}
$$

■

A.4 Miscellaneous Claims

In this section we prove some claims and corollaries that would be helpful to prove the rest of the properties.

Claim A.4.1 *Let $G_{n,p}$ be a random graph with $p \geq \frac{(\log n)^2}{n}$. Then*

$$\mathbb{P}[G_{n,p} \textit{ has non-large vertices }] = o\left(\frac{1}{\textit{poly}}\right)$$

Proof of Claim A.4.1: The property that all vertices are large is increasing, hence it is sufficient to prove for $p = \frac{(\log n)^2}{n}$. The expected degree of a vertex in $G_{n,p}$ is $(n-1)p$. Hence, using Chernoff bound (Eq. (A.1)), we get,

$$
\begin{aligned}
\mathbb{P}[G_{n,p} \text{ has non-large vertex}] &\leq n \cdot \mathbb{P}[v \text{ is non-large}] \leq n\mathbb{P}[\deg(v) \leq \tfrac{1}{2}(n-1)p] \\
&\leq 2n \cdot e^{-\Theta(\log^2(n))} \leq 2n \cdot n^{-\Theta(\log n)} = o\left(\frac{1}{\text{poly}}\right)
\end{aligned}
$$

■

Claim A.4.2 *Let $G_{n,p}$ be a random graph with $\frac{\log n}{n} \leq p \leq \frac{(\log n)^2}{n}$. Let $1 \leq \lambda \leq 10$ and $0 < \kappa \leq 10$ be two constants, and assume $\kappa - \lambda - 1 \geq 0.01$.*

Define $\mathcal{E}$ as the following event. For any two disjoint sets of vertices A,B, if $|A| = a \leq n^{\frac{1}{\log\log n}}$ and $|B| = \lambda a$, then $e(A,B) < \kappa|A|$. Then,

$$\mathbb{P}[\overline{\mathcal{E}}] \leq \begin{cases} o\left(\frac{1}{poly}\right) & \text{if } a = \omega(1) \\ n^{-0.9a(\kappa-(\lambda+1))} & \text{otherwise} \end{cases}$$

Proof of Claim A.4.2: Since $\mathcal{E}$ is a decreasing property, we prove for $p = \frac{(\log n)^2}{n}$. The expected number of edges between A and B is

$$\mathbb{E}(e(A,B)) = \frac{\lambda a^2(\log n)^2}{n}$$

For fixed sets A and B, we use Chernoff bound (Eq. (A.2)) with $\alpha = \frac{\kappa n}{\lambda a(\log n)^2}$. Then the probability that such two sets exist is at most

$$\begin{aligned} &\sum_a \binom{n}{a}\binom{n}{\lambda a}\left(\frac{\lambda e \cdot a(\log n)^2}{\kappa n}\right)^{\kappa a} \\ &\leq \sum_a \left[\frac{ne}{a}\left(\frac{ne}{a}\right)^{\lambda}\left(\frac{e\cdot a(\log n)^2}{n}\right)^{\kappa}\right]^a \\ &\leq \sum_a \left[\left(\frac{a}{n}\right)^{\kappa-(\lambda+1)} e^{\kappa}(\log n)^{2\kappa}\right]^a \end{aligned}$$

If $\omega(1) \leq a \leq n^{1/\log\log n}$ then since $\kappa - (\lambda+1) \geq 0.01$,

$$\leq n \cdot \left[n^{0.01\left(\frac{1}{\log\log n}-1\right)}(\log n)^{21}\right]^a \leq n \cdot n^{-0.005\omega(1)} = o\left(\frac{1}{\text{poly}}\right) \tag{A.5}$$

If $a \leq \log\log n$ then

$$\leq \log\log n \left[\left(\frac{\log\log n}{n}\right)^{\kappa-(\lambda+1)}(\log n)^{21}\right]^a \leq n^{-0.9a(\kappa-(\lambda+1))} \tag{A.6}$$

■

Claim A.4.3 *Let $0 < \beta \leq 3$ be a constant. Define $\mathcal{E}$ as the following event. For any set S of tiny vertices, if $|S| = s \leq n^{\frac{1}{\log\log n}}$ then $e(S) < \beta s$. Then,*

$$\mathbb{P}[\overline{\mathcal{E}}] \leq \begin{cases} o\left(\frac{1}{poly}\right) & \text{if } s = \omega(1) \\ n^{-0.9\beta s} & \text{otherwise} \end{cases}$$

Proof of Claim A.4.3: Using a similar argument as in Claim A.2.1, if S is a set of tiny vertices, then each vertex in S has at most $(\log\log n)^{c_2}$ neighbors in $V\backslash S$, and this is independent for every vertex in S. Moreover, this event and the event of having βs edges inside S are independent.

For a set S

$$\mathbb{E}(e(S)) = \tfrac{1}{2}s(s-1)p$$

Using Chernoff bound (Eq. (A.2)) with $\alpha = \frac{2\beta}{p(s-1)}$, gives us that the probability that such set exists is at most

$$\sum_s \binom{n}{s} \left[\sum_{i=0}^{(\log\log n)^{c_2}} \binom{n-s}{i} p^i (1-p)^{n-s-i} \right]^s \left(\frac{esp}{2\beta}\right)^{\beta s}$$

$$\leq \sum_s \left[\left(\frac{ne}{s}\right) (\log\log n)^{c_2} n^{(\log\log n)^{c_2}} p^{(\log\log n)^{c_2}} (1-p)^{n-s-(\log\log n)^{c_2}} \left(\frac{esp}{2\beta}\right)^{\beta} \right]^s$$

$$\leq \sum_s \left[\frac{ne}{s} \cdot (\log n)^{3(\log\log n)^{c_2}} e^{-\log n}(1+o(1)) \cdot \left(\frac{es(\log n)^2}{2n\beta}\right)^{\beta} \right]^s$$

If $\omega(1) = s \leq n^{\frac{1}{\log\log n}}$, then

$$\leq n \left[n^{\beta\left(\frac{1}{\log\log n}-1\right)} n^{\frac{\beta}{20}} \right]^s \leq n \cdot n^{-\frac{\beta\omega(1)}{4}} = o\left(\frac{1}{\text{poly}}\right)$$

If $s < \log\log n$, then

$$\leq \left(\frac{(\log n)^{5(\log\log n)^{c_2}}}{n^{\beta}}\right)^s \leq n^{-0.9\beta s}$$

■

Claims A.4.2 and A.4.3 imply the following corollary.

Corollary A.4.4 *Let $G_{n,p}$ be a random graph with $\frac{\log n}{n} \leq p \leq \frac{(\log n)^2}{n}$. For a vertex set S, let $\overline{d} = \frac{1}{|S|}\sum_{v\in S}\deg(v)$. Let $\eta \geq 0.1$ be a constant. Define $\mathcal{E}$ as the following event. For any set of tiny vertices S, $|N(S)| \geq (\overline{d} - (1+\eta))|S|$. Then,*

$$\mathbb{P}[\overline{\mathcal{E}}] = \begin{cases} o\left(\frac{1}{poly}\right) & \text{if } |S| = \omega(1) \\ 2n^{-0.3\eta|S|} & \text{otherwise} \end{cases}$$

Proof : Let $|S| = s$. Define $\mathcal{E}_1$ as the event $e(S) \leq \frac{\eta}{3}s$. Claim A.4.3 implies

$$\mathbb{P}[\overline{\mathcal{E}}_1] \leq \begin{cases} o\left(\frac{1}{\text{poly}}\right) & \text{if } s = \omega(1) \\ n^{-0.3\eta s} & \text{otherwise} \end{cases}$$

If $\mathcal{E}_1$ holds, then $e(S, N(S)) \geq (\overline{d} - \frac{2}{3}\eta)s$. Let $\mathcal{E}_2$ be the event $|N(S)| \geq (\overline{d} - (\eta+1))s$. If $\mathcal{E}_1$ holds and $\mathcal{E}_2$ does not hold, then we can apply Claim A.4.2 with $\kappa \geq (\overline{d} - \frac{2}{3}\eta)$ and $\lambda \leq \overline{d} - (1+\eta)$. Since in this case

$$\kappa - \lambda \geq -\frac{2}{3}\eta + 1 + \eta = 1 + \frac{\eta}{3} \geq 1.01$$

then,

$$\mathbb{P}[\overline{\mathcal{E}}_2|\mathcal{E}_1] \leq \begin{cases} o\left(\frac{1}{\text{poly}}\right) & \text{if } s = \omega(1) \\ n^{-0.3\eta s} & \text{otherwise} \end{cases}$$

Since $\mathbb{P}[\overline{\mathcal{E}}_2] \leq \mathbb{P}[\overline{\mathcal{E}}_2|\mathcal{E}_1] + \mathbb{P}[\mathcal{E}_1]$ this completes the proof. ■

Claim A.4.5 *Let $G_{n,p}$ be a random graph with $\frac{\log n}{n} \leq p \leq \frac{(\log n)^2}{n}$. Define $\mathcal{E}$ as the following event.*

- *There are no two vertices of degree at most 5, and their distance is 4 or less. And,*
- *There is no vertex of degree at most 5, who belongs to a cycle of length 4 or less.*

Then,

$$\mathbb{P}[\overline{\mathcal{E}}] \leq 2\frac{(\log n)^{14}}{n}$$

Proof of Claim A.4.5: Let $d' = np$. Then $\log n < d' < (\log n)^2$. The probability for the first event is at most

$$\begin{aligned} &\binom{n}{2}\left[\sum_{i=0}^{5}\binom{n-2}{i}p^i(1-p)^{n-2-i}\right]^2 \cdot (p + np^2 + n^2p^3 + n^3p^4) \\ &\leq \frac{n^2}{2}5\frac{n^{10}}{5!5!} \cdot p^{10} \cdot e^{-2np}(1+o(1)) \cdot 2n^3p^4 \leq \frac{1}{5!}d'^{14}ne^{-2d'} \leq \frac{(\log n)^{14}}{n} \end{aligned}$$

The probability for the second event is at most

$$\begin{aligned} &n\sum_{i=0}^{5}\binom{n-1}{i}p^i(1-p)^{n-1-i} \cdot (np^2 + n^2p^3 + n^3p^4) \\ &\leq n \cdot 5 \cdot \frac{n^5}{5!} \cdot p^5 \cdot e^{-np}(1+o(1)) \cdot 2n^3p^4 \leq \frac{1}{10}(d')^{10} \cdot e^{-d'} \leq \frac{(\log n)^{10}}{n} \end{aligned}$$

■

Claim A.4.5 leads to the following corollary.

Corollary A.4.6 *Let S be a set of vertices. Divide S into $S_1 = \{v \in S : \deg(v) \leq 5\}$ and $S_2 = S \backslash S_1$. Then,*

(a) $N(S_1) = \sum_v \deg(v)$.

(b) $|N(S)| \geq |N(S_1)| + |N(S_2)| - |S_2| - e(S)$.

(c) $|N(S)| \geq |N(S_1)| + |N(S_2)| - 2|S_2|$

Proof of Corollary A.4.6: For (a), if $|N(S)| < \sum_v \deg(v)$ then either there are two vertices in S_1 with edge between them or they have a common neighbor. But in these cases the distance between these vertices will be 1 or 2.

For (b) and (c) observe that

$$|N(S)| = |N(S_1)| + |N(S_2)| - |N^*(S_1) \cap N(S_2)| - |N(S_1) \cap S_2|$$

if $|N^*(S_1) \cap N(S_2)| > |S_2|$ then it means that there are vertices $v_1, v_2 \in N^*(S_1)$, and a vertex $w \in S_2$, such that $v_1, v_2 \in N(w)$. If v_1, v_2 in S_1 then their distance is 2. If $v_1, v_2 \in N(S_1)$ then let v_1', v_2' be their neighbors in S_1. Then v_1' and v_2' are in distance of 4, or if $v_1' = v_2'$ then it is on a cycle of length 4. W.l.o.g., if $v_1 \in S_1$ and $v_2 \in N(S_1)$ then let v_2' be the neighbor of $v_2 \in S_1$. Then v_1 and v_2' are in distance of 3, of if $v_1 = v_2'$ it is on a cycle of length 3.

$|N(S_1) \cap S_2|$ is bounded by $|S_2|$ which proves (c). But it is also bounded by $e(S_1, S_2) \le e(S_1 \cup S_2) = e(S)$ which proves (b). ■

Claim A.4.7 *Let $G_{n,p}$ be a random graph with $p \le \frac{(\log n)^2}{n}$. Let $1.1 \le \gamma \le 3$ be a constant. Define $\mathcal{E}$ as the following event. Any set S, $|S| = s \le \frac{n}{(\log n)^{100}}$ spans at most γs edges. then*

$$\mathbb{P}[\overline{\mathcal{E}}] = \begin{cases} o\left(\frac{1}{poly}\right) & \text{if } |S| = \omega(1) \\ n^{-0.9s(\gamma-1)} & \text{otherwise} \end{cases}$$

Proof of Claim A.4.7: $\mathcal{E}$ is an increasing property, hence we use $p = \frac{(\log n)^2}{n}$. Let S be a subset of vertices, $|S| = s \le \frac{n}{(\log n)^5}$. Then $\mathbb{E}(e(S)) = \binom{s}{2} \cdot p = \frac{s(s-1)(\log n)^2}{2n}$. Using Eq. (A.2), taking $\alpha = \frac{2\gamma n}{(s-1)(\log n)^2}$, we have that for a fixed set S,

$$\mathbb{P}[e(S) \ge \gamma|S|] \le \left(\frac{e}{\alpha}\right)^{\alpha \mathbb{E}(e(S))} e^{-\mathbb{E}(e(S))} \le \left(\frac{e(s-1)(\log n)^2}{2\gamma n}\right)^{\gamma s}$$

Using union bound, the probability that such a set exist is at most,

$$\sum_{1}^{n/(\log n)^{100}} \binom{n}{s} \left(\frac{es(\log n)^2}{2\gamma n}\right)^{\gamma s} \le \sum_{1}^{n/(\log n)^{100}} \left[\frac{ne}{s}\left(\frac{es(\log n)^2}{2\gamma n}\right)^{\gamma}\right]^s$$

$$\le \sum_{1}^{n/(\log n)^{100}} \left[\left(\frac{s}{n}\right)^{\gamma-1} \cdot (\log n)^{2\gamma+1}\right]^s$$

If $\Omega(\log n) = s \le \frac{n}{(\log n)^{100}}$, then $\frac{s}{n} \le (\log n)^{-100}$ and

$$\le n\left((\log n)^{-100(\gamma-1)+2\gamma}\right)^s \le n\left(\frac{1}{\log n}\right)^{\Omega(\log n)} = o\left(\frac{1}{\text{poly}}\right)$$

If $s \leq \log n$, then $\frac{s}{n} \leq \frac{\log n}{n}$ and

$$\leq \sum_{s} \left(\frac{(\log n)^{3\gamma}}{n^{\gamma-1}} \right)^{s} \leq n^{0.9s(\gamma-1)} \tag{A.7}$$

which is $o\left(\frac{1}{\text{poly}}\right)$ if $s = \omega(1)$, and therefore completes the proof. ■

A.5 Expansion Properties : P7 - P10

A.5.1 Proof of Property P7

Claim A.5.1 *Let $G_{n,p}$ be a random graph with $p \geq \frac{\log n}{n}$, then*

$$\mathbb{P}[G_{n,p} \notin P7] = o\left(\frac{1}{poly}\right)$$

Proof of Claim A.5.1: Let $|S|$ be a set of large vertices, $|S| = s$. We prove in two stages. We first prove for the case that $n^{0.1} \leq s \leq \frac{n}{\log n}$. Since this property is increasing it is enough to prove for $p = \frac{\log n}{n}$.

Let $\alpha = \frac{1}{300}$. For a vertex $v \in V \backslash S$, the probability that v has a neighbor in S is $p_1 = 1 - (1-p)^s$. Using Taylor expansion, $ps - \frac{1}{2}p^2s^2 \leq p_1 \leq ps$. Let $q_1 = 1 - p_1 = (1-p)^s \leq e^{-ps}$. Observe that the event of having a neighbor in S is independent for each $v \in V \backslash S$. Hence, the size of $N(S)$ is distributed as $B(n-s, p_1)$. Moreover, $\mathbb{E}(B(n-s,p_1)) = (n-s)p_1 \geq \alpha s \log n$ (see below). Hence, for a fixed set S,

$$\begin{aligned}
\mathbb{P}[|N(S)| \leq \alpha s \log n] &\leq \sum_{t=0}^{\alpha s \log n} \binom{n-s}{t} {p_1}^t {q_1}^{n-s-t} \\
&\leq \alpha s \log n \binom{n-s}{\alpha s \log n} {p_1}^{\alpha s \log n} {q_1}^{n-s-\alpha s \log n} \\
&\leq n \binom{n-s}{\alpha s \log n} (ps)^{\alpha s \log n} e^{-ps(n-s-\alpha s \log n)} \\
&\leq n \left(\frac{neps}{\alpha s \log n} \right)^{\alpha s \log n} e^{-s \log n} e^{s} e^{\alpha s \log n} \\
&\leq n n^{-s} \left(\frac{e^3}{\alpha \log n} \right)^{\alpha s \log n} \leq n n^{-s} (0.5 \log n)^{-\alpha s \log n}
\end{aligned}$$

using $s \log n \leq n$, $ps \leq 1$ and $pn = \log n$.

$$\begin{aligned}
\mathbb{P}[\exists S\ , |N(S)| \leq \alpha \log n s] &\leq \sum_{s=n^{0.1}}^{n/\log n} \binom{n}{s} \mathbb{P}[|N(S)| \leq \alpha s \log n] \\
&\leq n^2 n^s \cdot n^{-s} \cdot (0.5 \log n)^{-\alpha s \log n} \\
&\leq n^2 (0.5 \log n)^{-\alpha s \log n} = o\left(\frac{1}{\text{poly}}\right)
\end{aligned}$$

To complete the proof for $s > n^{0.1}$, we need to prove that $(n-s)p_1 \geq \alpha s \log n$. Since $p_1 \geq sp - \frac{1}{2}s^2p^2$, it is enough to prove $(n-s)(sp - \frac{1}{2}s^2p^2) \geq \alpha s \log n$, which is the same as $(n-s)(p - \frac{1}{2}sp^2) \geq \alpha \log n$. Continue with the left hand side,

$$\begin{aligned}(n-s)\left(p - \tfrac{1}{2}sp^2\right) &\geq \left(n - \frac{n}{\log n}\right)\left(p - \tfrac{1}{2}\frac{n}{\log n}p^2\right) \\ &= \left(n - \frac{n}{\log n}\right)\tfrac{1}{2}p = \tfrac{1}{2}\log n - \tfrac{1}{2} \geq \alpha \log n.\end{aligned}$$

We turn to the second stage of the proof, where $s \leq n^{0.1}$. We first consider the case that $\frac{\log n}{n} \leq p \leq \frac{(\log n)^2}{n}$. Assume that $|N(S)| \leq \frac{\log n}{300}|S|$, then $|N^*(S)| \leq \left(1 + \frac{\log n}{300}\right)|S|$ and therefore,

$$|S| \geq \frac{|N^*(S)|}{1 + \frac{\log n}{300}} \geq \frac{200\,|N^*(S)|}{\log n}\ , \tag{A.8}$$

then,

$$e(N^*(S)) \geq \frac{\log n}{20}|S| \geq 10\,|N^*(S)| \tag{A.9}$$

Since $p \leq \frac{(\log n)^2}{n}$ and $|N^*(S)| \leq \frac{n}{(\log n)^5}$ then Claim A.4.7 implies that the probability that Eq. A.9 is valid is $o\left(\frac{1}{\mathrm{poly}}\right)$.

For $p \geq \frac{(\log n)^2}{n}$, define the property P7' as P7 and $G_{n,p}$ contains only large vertices. This property is increasing, and hence it is enough to prove for $p = \frac{(\log n)^2}{n}$. Using Claim A.4.1,

$$\mathbb{P}[G_{n,p} \notin P7'] \leq \mathbb{P}[G_{n,p} \notin P7] + \mathbb{P}[G_{n,p} \text{ has non-large vertex }] = o\left(\frac{1}{\mathrm{poly}}\right)$$

Since $\mathbb{P}[G_{n,p} \in P7] \geq \mathbb{P}[G_{n,p} \in P7']$ this completes the proof. ■

Before proving the rest of the expansion properties, we remark that Claim A.4.1 implies that if $p \geq \frac{(\log n)^2}{n}$, then $P7$ implies $P8 - P10$. Hence, to the rest of this section we assume $p \leq \frac{(\log n)^2}{n}$.

A.5.2 Proof of Property P8

Claim A.5.2 *Let $G_{n,p}$ be a random graph with $p \geq \frac{\log n}{n}$, then*

$$\mathbb{P}[G_{n,p} \notin P8] = o\left(\frac{1}{poly}\right)$$

Proof of Claim A.5.2: Let S be a set of small vertices. If P8 does not hold, then $|N(S)| \leq (\log\log n)^{c_2-1}|S|$ and hence $|N^*(S)| \leq \left(1+(\log\log n)^{c_2-1}\right)|S|$. Therefore,

$$|S| \geq \frac{|N^*(S)|}{1+(\log\log n)^{c_2-1}} \geq \frac{|N^*(S)|}{2(\log\log n)^{c_2-1}}\ , \tag{A.10}$$

and

$$e(N^*(S)) \geq \tfrac{1}{2}(\log\log n)^{c_2}\cdot|S| \geq \tfrac{1}{4}\log\log n|N^*(S)| \geq 3|N^*S| \tag{A.11}$$

Since $(\log\log n)^{c_2} \leq |N^*(S)| \leq \frac{n}{(\log n)^{100}}$ then Claim A.4.7 completes the proof. ■

A.5.3 Proof of Property P9

Claim A.5.3 *Let $G_{n,p}$ be a random graph with $p \geq \frac{\log n}{n}$. Define $\mathcal{E}$ as the following event. For any set S of tiny vertices, if $|S| \geq \log\log n$ and at most $|S|/4$ of the vertices in S are of degree 2, then $|N(S)| \geq \frac{9}{4}|S|$. Then,*

$$\begin{aligned}\mathbb{P}[G_{n,p} \notin P9] &\leq \mathbb{P}[\overline{\mathcal{E}}]\\ &\leq \begin{cases} (a) & o\left(\frac{1}{poly}\right) \quad \textit{if the minimum degree in } S \textit{ is } 4\\ (b) & \frac{polylog(n)}{n} \quad \textit{otherwise}\end{cases}\end{aligned}$$

Proof of Claim A.5.3: Let S be a set of tiny vertices, $|S| = s \geq \log\log n$. Define $\overline{d}$ as the average degree of the vertices in S. If $\overline{d} \geq 3.5$ then using Corollary A.4.4 with $\eta = 0.1$ implies (a). If $d \leq 3.5$ then at least half of the vertices in S are of degree 4 or less. Let S_1 be the set of these vertices and let $S_2 = S\backslash S_1$, $s_1 = |S_1|$ and $s_2 = |S_2|$.

Since $s_1 \geq 0.5s$ and since there are at most $0.25s$ vertices of degree 2, then Corollary A.4.6 implies that with probability $1 - \frac{\mathrm{polylog}(n)}{n}$

$$|N(S_1)| \geq 2.5s_1 \tag{A.12}$$

Consider s_2. If $s_2 \leq 0.01s$ then

$$|N(S)| \geq |N(S_1)| - s_2 \geq 2.5\cdot 0.99s - 0.01s \geq 2.25s \tag{A.13}$$

and (b) follows.

If $s_2 > 0.01s = \omega(1)$, then since the minimum degree in s_2 is 5, then using Corollary A.4.4 with $\eta = 0.1$, implies that with probability $1 - o\left(\frac{1}{\mathrm{poly}}\right)$

$$N(S_2) \geq 3.5|S_2| \tag{A.14}$$

Claim A.4.3 implies that with probability $1 - o\left(\frac{1}{\mathrm{poly}}\right)$

$$e(S) \leq \frac{1}{100}|S| \tag{A.15}$$

Combining Eq. (A.12), (A.14) and (A.15), and using Corollary A.4.6(b) we get that with probability $1 - \frac{\mathrm{polylog}(n)}{n}$

$$N(S) \geq 2.5s_1 + 3.5s_2 - s_2 - 0.01s \geq 2.25s$$

which completes the proof of (b). ■

A.5.4 Proof of Property P10

Claim A.5.4 *Let $G_{n,p}$ be a random graph with $p \geq \frac{\log n}{n}$.*

$$\mathbb{P}[G_{n,p} \notin P10] = \mathbb{P}[\exists S, |S| \leq \frac{n}{\log n}, |N(S)| < 2|S|] =$$

$$\leq \begin{cases} (a) \quad o\left(\frac{1}{poly}\right) & \textit{if } S \textit{ has no tiny vertices} \\ (b) \quad o\left(\frac{1}{poly}\right) & \textit{if the minimum degree in } S \textit{ is } 5 \textit{ and } |S| = \omega(1) \\ (c) \quad o\left(\frac{1}{n^{0.5|S|}}\right) & \textit{if the minimum degree in } S \textit{ is } 5 \textit{ and } |S| \geq 2 \\ (d) \quad \frac{polylog(n)}{n} & \textit{otherwise} \end{cases}$$

Proof of Claim A.5.4: Consider a set S, $|S| = s$. First assume that S consists of small and large vertices. If $|S| \geq n^{0.1}$ then the first stage in the proof of Claim A.5.1 implies (a) (observe that the proof there is for any set of vertices). If $s \leq n^{0.1}$ then the same proof as in Claim A.5.2 implies (a).

To prove (b) and (c) we divide S into two sets: $S_1 = S \cap \mathrm{TINY}$, and $S_2 = S \backslash S_1$, $s_1 = |S_1|$ and $s_2 = |S_2|$. If $s_2 \geq 0.01s$ then Claim A.5.2 implies that with probability $1 - o\left(\frac{1}{\mathrm{poly}}\right)$,

$$|N(S)| \geq |N(S_2)| - |S_1| \geq 0.01(\log\log n)^{c_2 - 1}s - 0.99s \geq 10s\ . \tag{A.16}$$

Otherwise $s_1 > 0.99s$. In this case if $|N(S_1)| \geq 2.25s_1$ then

$$|N(S)| \geq |N(S_1)| - |S_2| \geq 2.25 \cdot 0.99s - 0.01s \geq 2s\ .$$

Using Corollary A.4.4 with $\eta = 1.75$, implies that

$$\mathbb{P}[N(S_1) \geq 2.25|S_1|] \leq \begin{cases} o\left(\frac{1}{\mathrm{poly}}\right) & \text{if } |S_1| = \omega(1) \\ n^{-0.525s} & \text{otherwise} \end{cases}\ ,$$

which complete the proof of (b) and (c).

To prove (d) we divide S into three sets. S_1 is the set of vertices of degree 5 or less, S_2 is the set of tiny vertices with degree 6 or more, and S_3 is the rest of the vertices. Let s_1,s_2,s_3 be their sizes.

Since the minimum degree is 2, then Claim A.4.5 implies

$$\mathbb{P}[|N(S_1)| \geq 2|S_1|] \geq 1 - \frac{\mathrm{polylog}(n)}{n} \tag{A.17}$$

Consider the sets S_2 and S_3. Next we prove that

$$\mathbb{P}[|N(S_2 \cup S_3)| \geq 4(s_2 + s_3)] \geq 1 - o\left(\frac{1}{n}\right) \tag{A.18}$$

If $s_3 \geq (s_2 + s_3)/100$, then using the same argument as in Eq. (A.16) implies Eq. (A.18).

If $s_2 \geq 0.99(s_2 + s_3)$, and $s_2 \geq 4$, then using Corollary A.4.4 with $\eta = 0.9$,

$$\mathbb{P}[N(S_2) \leq 4.1 s_2] \leq n^{-0.3 \cdot 0.9 s_2} = o\left(\frac{1}{n}\right).$$

This implies that with probability $1 - o\left(\frac{1}{n}\right)$

$$|N(S_2 \cup S_3)| \geq |N(S_2)| - |S_3| \geq 4.1 \cdot 0.99(s_2 + s_3) - 0.01(s_2 + s_3) \geq 4s$$

The last possibility is when $s_2 \leq 3$ and in this case $s_3 = 0$. If $s_2 = 1$, then (A.18) must hold.

If $s_2 = 2$ and $N(S_2) \leq 7$ then $N^*(S_2) \leq 9$. In this case, $e(N^*(S_2)) \geq 11$ then using Claim A.4.7 the probability for this is at most $n^{-0.9 \cdot 9 \cdot 2/9} \leq o\left(\frac{1}{n}\right)$.

If $s_2 = 3$, then using Claim A.4.3 with $\beta = 0.5$ implies that the probability that there are two edges inside S_2 is $o\left(\frac{1}{n}\right)$. If this holds and $|N(S_2)| \leq 11$, then $N^*(S)$ is of size 14 and spans 17 edges. Then according to Claim A.4.7 the probability for this is at most $n^{-0.9 \cdot 14 \cdot (3/14)} = o\left(\frac{1}{n}\right)$.

This completes the proof of Eq. (A.18). Then combining Corollary A.4.6(c), Eq. (A.17) and Eq. (A.18) completes the proof of the claim. ■

A.6 Proof of Property P11

Claim A.6.1 *Let $G_{n,p}$ be a random graph with $p \leq \frac{\log n}{n}$, then*

$$\mathbb{P}[G_{n,p} \notin P11] = o\left(\frac{1}{poly}\right)$$

Proof of Claim A.6.1: We use $p = \frac{\log n}{n}$. Recall that this property claims that if A and B are two disjoint sets of vertices, such that $|A|, |B| \leq \frac{30 n \log\log n}{\log n}$, then $e(A, B) \leq 300(\log\log n)\max\{|A|, |B|\}$. Since this property is decreasing, we assume w.l.o.g. that $|A| = |B| = a$.

$$\begin{aligned}
\mathbb{P}[G_{n,p} \notin P11] &\leq \sum_a \binom{n}{a}\binom{n}{a}\binom{a^2}{300a\log\log n} p^{300a\log\log n} \\
&\leq \sum_a \left[\frac{ne}{a}\frac{ne}{a}\left(\frac{a^2 e}{300a\log\log n}\right)^{300\log\log n}\left(\frac{\log n}{n}\right)^{300\log\log n}\right]^a \\
&\leq \sum_a \left[\frac{ne}{a}\left(\frac{e\log n}{300\log\log n}\cdot\frac{a}{n}\right)^{150\log\log n}\right]^{2a}
\end{aligned}$$

If $a \geq \frac{n}{(\log n)^5}$, then in this case $\frac{n}{a} \leq (\log n)^5$ and $\frac{a}{n} \leq \frac{30 \log\log n}{\log n}$

$$\leq \sum_a \left[(\log n)^6 \left(\frac{e \log n}{300 \log\log n} \cdot \frac{30 \log\log n}{\log n} \right)^{150 \log\log n} \right]^{2a}$$

$$\leq n \left[(\log n)^6 e^{-150 \log\log n} \right]^a \leq (\log n)^{-100a} = o\left(\frac{1}{\text{poly}}\right)$$

If $\log n \leq a \leq \frac{n}{(\log n)^5}$, then in this case $\frac{a}{n} \leq \frac{1}{(\log n)^5}$ and

$$\leq \sum_a \left[\left(\frac{a}{n}\right)^{150 \log\log n - 1} (\log n)^{150 \log\log n} \right]^{2a}$$

$$\leq (\log n)^{-500 \log\log n \cdot \log n} = o\left(\frac{1}{\text{poly}}\right)$$

If $a \leq \log n$, then in this case $\frac{a}{n} \leq \frac{\log n}{n}$ and

$$\leq \log n \left(\frac{(\log n)^3}{n} \right)^{150 \log\log n} = o\left(\frac{1}{\text{poly}}\right)$$

■

Appendix B

Proof of Claim 6.5.3

B.1 Proof of the Size of $\hat{A}$ and $\hat{B}(a)$

We have αn^2 pairs (a, b). We divide the a-s into two sets. The first set is $\hat{A}$, where each $a \in \hat{A}$ appears in at least $\alpha n/2$ pairs. The second set is $\overline{\hat{A}}$ where each a in this set appears in at most $\alpha n/2$ pairs. Hence :

$$\#\text{ of pairs } \leq \left|\hat{A}\right| \cdot n + \left|\overline{\hat{A}}\right| \cdot \alpha n/2 ,$$

since $\left|\overline{\hat{A}}\right| \leq n$, we get

$$\alpha n^2 \leq \left|\hat{A}\right| \cdot n + n \cdot \alpha n/2 = \left|\hat{A}\right| \cdot n + \alpha n^2/2 ,$$

which implies,

$$\left|\hat{A}\right| \cdot n \geq \alpha n^2 - \alpha n^2/2 = \alpha n^2/2 ,$$

hence,

$$\left|\hat{A}\right| \geq \alpha n/2$$

The size of $\hat{B}(a)$ is true by definition.

B.2 Lower bound for $\frac{\alpha n}{2}$

We need to prove that $2\alpha^{-1} \leq \frac{\log n}{20 \log\log n}$. Since $2\alpha^{-1} = 4\mu^2 k! 6^k \leq \mu^2 (6k)^k$, and μ is constant, then it is enough to prove that if c_4 is large constant, then for n

large enough, $(6k)^k \leq \frac{\log n}{c_4 \log\log n}$.

$$
\begin{aligned}
(6k)^k &= \left(\frac{\log\log n}{\log\log\log n}\right)^{\frac{\log\log n}{6\log\log\log n}} \\
&= \left(\left(\frac{\log\log n}{\log\log\log n}\right)^{\frac{\log\log n}{\log\log\log n}}\right)^{1/6} \\
&= \left(\frac{\log n}{(\log\log\log n)^{\frac{\log\log n}{\log\log\log n}}}\right)^{1/6} \\
&< \left(\frac{\log n}{\log\log n}\right)^{1/6} \\
&\leq \frac{\log n}{c_4 \log\log n}
\end{aligned}
$$