Finding hidden Hamiltonian cycles

Andrei Z. Broder* Alan M. Frieze† Eli Shamir‡

Abstract

Consider a random graph G composed of a Hamiltonian cycle on n labeled vertices and dn random edges that "hide" the cycle. Is it possible to unravel the structure, that is, to efficiently find a Hamiltonian cycle in G?

We describe an $O(n^3 \log n)$ steps algorithm A for this purpose, and prove that it succeeds almost surely. Part one of A properly covers the "trouble spots" of G by a collection of disjoint paths. (This is the hard part to analyze.) Part two of A extends this cover to a full cycle by the rotation-extension technique which is already classical for such problems.

1 Introduction

Graph theoretic algorithms typically involve a search for a subgraph of the input graph that satisfies certain given properties. Often the associated decision problem (i.e. deciding whether such subgraph exists) is NP-hard although the search problem is easy most of the the time if the input graph is

*DEC Systems Research Center, 130 Lytton Ave, Palo Alto, CA.

†Department of Mathematics, Carnegie-Mellon University. A portion of this work was done while the author was visiting DEC SRC. Supported in part by NSF grant CCR 0890012.

‡The Institute of Mathematics and Computer Science, Hebrew University, Jerusalem. A portion of this work was done while the author was visiting DEC SRC. Supported in part by grant 438/89 of the Israeli Academy of Sciences.

chosen according to some natural probability distribution. This clearly depends both on the given distribution and on the ingenuity of the algorithm.

In this article we study the "hidden structure" version of such a search problem, that is, we are given a graph G that is known to contain a Hamiltonian cycle H, "hidden" among a relatively few additional random edges that by themselves are not likely to induce a Hamiltonian cycle. Our goal is to find some Hamiltonian cycle in G, not necessarily H. A preliminary version of this paper has appeared in [4].

More precisely, a random graph $G = (V, E)$ in this problem is defined by n labeled vertices and the following set of edges:
(i) The n edges forming a specific Hamiltonian cycle.
(ii) The edges obtained by choosing each pair of distinct vertices $\{i, j\} \subset V$ to be an edge with probability d/n, independently for all pairs.
We assume that d is a constant. (Our algorithm extends easily for the case when d is a growing function of n. We omit the details here.)

Condition (ii) alone defines the Erdös-Renyi space $G(n, p)$, and for the given $p = d/n$, the graphs in this space are typically sparse with about $dn/2$ edges. The class of graphs in $G(n, d/n)$ that are Hamiltonian has a negligible probability as $n \to \infty$.

Condition (i) means that we apply a "magnifying glass" to boost this probability to 1, and now ask for an efficient search algorithm that that will a.s. (almost surely) produce *some* Hamiltonian cycle in G. The literature of recent years abounds with search algorithms for Hamiltonian cycles in random graphs ([1, 2, 3, 9, 10, 11, 17]), but all the prior methods work only for much denser graphs or sparse but regular graphs, and are not directly useful for our problem. We present a new algorithm, which finds a Hamiltonian cycle a.s., in the more demanding situation when the input space is defined by (i) and (ii).

There are several motivations for the hidden structure algorithms:

1. *Customized algorithms.* Usually the search algorithms for random graphs rely heavily on the statistical properties of the input space and to a great extent proceed locally from step to step with no regard to the global input. If one makes more stringent demands (e.g. very small failure probability, or small expected time) then a correct algorithm must scrutinize individual

inputs more carefully. The hidden structure algorithm for Hamiltonian cycle goes to the extreme in this respect; its success depends on a careful initial handling of all "trouble corners" of the input graph. Practical experience shows that this is a very good heuristic.

2. *Practical considerations.* Several hidden structure problems were studied before: minimum bandwidth [18], minimum bisection [5, 6, 7], maximum clique or maximum independent set [12], k-coloring, bisection-width, graph-partitioning, 3-partition [7]. One reason for this interest is that graphs that arise in practical applications (say VLSI design) tend to have a hidden structure (e.g. small bisection) that random graphs of the same density do not possess. Hence random graphs with hidden structure are more suitable models for studying expected or almost sure behavior of algorithms for hard search problems.

3. *Cryptography.* Modern cryptography is based on the concept of one-way functions, which are functions easy to compute (encryption) but hard to invert (decryption) on most instances. Common constructions for one-way functions are based on the conjectured difficulty of certain number-theoretical questions. The ability to invert a one-way function often constitute the essence of an authentication scheme.

It is natural to try to use an NP-hard problem for the same purpose. The encryption mechanism in this case would be to construct a random instance of an NP-hard problem with a known solution. The authentication protocol would be to present a solution to the given instance. For this scheme to be secure, the generated random instances should be hard on average. (Our discussion of cryptography here is necessarily brief and superficial. The interested reader should consult the rapidly mounting literature in this field.)

In particular, to use the Hamiltonian cycle problem for this purpose, one needs a probabilistic, polynomial-time algorithm for generating a hard-on-average distribution of solved instances; that is, the encryptor must be able to generate random (G, H) pairs in such a way that an adversary, upon seeing G, almost surely cannot compute H (or any other Hamiltonian cycle H' in G) in polynomial time.

A natural scheme to try is the one studied in this paper: Pick a random Hamiltonian cycle on n vertices, and add to it random edges, chosen independently with probability of existence d/n. (Observe that if $d > \ln n$

then almost surely G contains a Hamiltonian cycle made exclusively out of random edges and this cycle can be found in polynomial time [3].) Our algorithm shows that this generation scheme will not work for d greater than some constant: anyone who sees the graph G can, with high probability, find a Hamiltonian cycle H' in polynomial time. This adds to the rising evidence that combinatorial hidden structures (unlike the number theoretic ones) are much more prone to fail in a cryptographic sense, that is, they are easy to unravel.

1.1 Informal description of the algorithm

The approach behind our algorithm can be described as follows: if all vertices had large degree then it would be easy to prove that the graph G almost surely (*a.s.*) had a Hamilton cycle. We avoid the problem caused by the set of vertices X_0 of "low" degree by finding a collection of vertex disjoint paths P such that each $x \in X_0$ is an *internal* vertex of a path in P. We can then "shrink" the paths P to (required) edges. All vertices in $V \setminus X_0$ have "high" degree and the problem is solved. Unfortunately, shrinking P implies the deletion of edges incident with X_0, causing some vertices in $V \setminus X_0$ to become of low degree. We avoid this by replacing X_0 by a slightly larger set X which has the property that any vertex not in X has few neighbours not in X. X and P are constructed in Phases 1 and 2. (See [9] for a similar construction.)

The problem has now been reduced in essence to finding a Hamilton cycle in a random graph with vertices of high degree but only a linear number of random edges. We now use a version of the extension-rotation algorithm in which some (randomly chosen) *green* edges are only used for extensions. This strange artifact is needed in the proof of correctness of the algorithm (see also [9, 10]). We strongly suspect that this "trick" is unnecessary, but we cannot at present do without it.

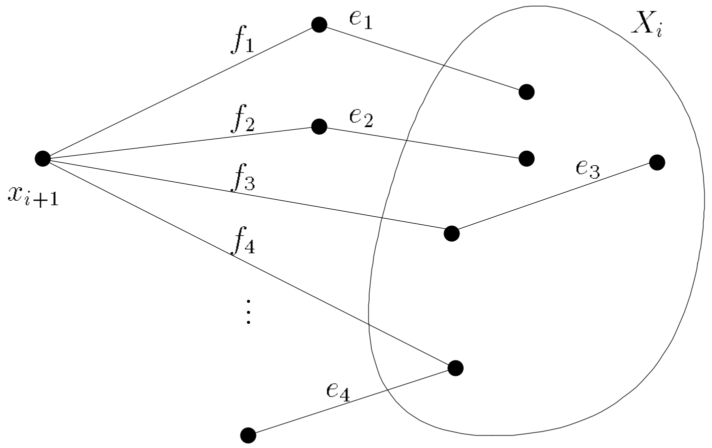

Figure 1: Construction of X.

2 The algorithm

We assume that the input consists of graph $G = (V, E)$ which is built from a Hamiltonian cycle H plus a random graph drawn from the distribution $G_{n,p}$, with $p = d/n$, with d greater than some large constant d_0.

The algorithm is divided into 4 phases.

Phase 1. Let K be a fixed large integer. (For instance, $K = 100$ will do.) Let $X_0 = \{v \in V \mid d(v) \leq d/4\}$, where $d(v)$ is the degree of v in G. Define $X_i = X_0 \cup \{x_1, x_2, \ldots, x_i\}$ iteratively by choosing x_{i+1} to be any vertex for which there are K *independent* (no common endpoints) edges, $e_1, \ldots, e_K$ and K other edges, $f_1, \ldots, f_K$ such that for $j = 1, 2, \ldots, K$

$$\begin{aligned} e_j \cap X_i &\neq \emptyset \\ |e_j \cap f_j| &= 1 \\ x_{i+1} &\in f_j \end{aligned} \tag{1}$$

(see Figure 1.)

Let $X = X(G, K)$ be the final set of vertices produced by this subroutine. Note that X does not depend on the order in which vertices are added since once a vertex becomes eligible for addition to X_i it remains eligible for addition to $X_{i+1}, X_{i+2}, \ldots$, until actually added.

Phase 2. Let E_X be the set of edges incident to X, in other words $E_X = \{e \in E \mid e \cap X \neq \emptyset\}$. Let G_X be the graph with edge set E_X. (We shall prove that G_X is a.s. composed only of trees and unicyclic graphs.)

Construct a set of vertex disjoint paths $\mathcal{P} = \{P_1, P_2, \ldots\}$ such that
(i) If $x \in X$ then x is an *internal* vertex of one of the P_i.
(ii) All paths are fully contained in E_X.
(iii) For any two consecutive vertices on a path, at least one vertex is in X. Thus all paths have endpoints *not* in X.

We shall show that given the above of structure of G_X and the fact that G is Hamiltonian, this construction succeeds in quadratic time.

Let Y denote the set of internal vertices on the paths in $\mathcal{P}$. We shall see later (see Remark 2) that a.s. every $v \notin Y$ has at most K neighbors in Y.

Delete all edges in E_X from G, except for the edges belonging to any path P_i. Denote the resulting graph G'. We shall show that G' is connected, and furthermore all vertices in $G' \setminus Y$ have degree at least $d/4 - K \geq d/5$.

(The final Hamiltonian cycle produced by our algorithm is contained in G' and will include all the paths found in this stage, unbroken.)

Phase 3. Color each edge $e \in E(G')$ randomly green or blue with equal probability. If a vertex $v \in G'$ has less than $d/4$ blue edges incident to it, recolor all the edges incident to it blue. Let $G_b = (V, E_b)$ be the graph constructed from the blue edges and let E_g be the set of green edges.

Phase 4. Now we find a Hamiltonian cycle, by constructing an increasingly longer path in stages.

Suppose that in stage r we have a path P of length r such that, for every $P_i \in \mathcal{P}$ either
(i) P_i is a subpath of P,
or
(ii) P_i is disjoint from P.

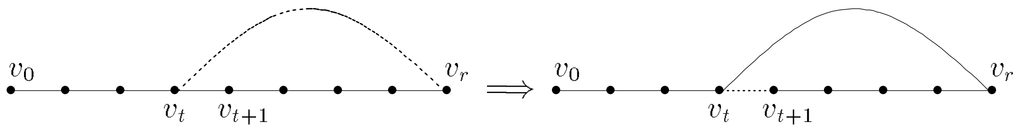

Figure 2: Rotations

A rotation in $P = (v_0, v_1, \ldots, v_r)$ consists in adding an edge $\{v_r, v_t\}$ and removing the edge $\{v_t, v_{t+1}\}$. See Figure 2.

Stage r **part 1.** Let v_0 be one end of P. Keeping v_0 fixed do rotations in an arbitrary order under the following restrictions:

a. Only blue edges can be used as rotations edges.

b. An edge can be used at most once as a rotation edge.

c. A vertex other than v_0, can become an endpoint at most once.

d. The removed edge, is not an edge of some P_i.

until one out of the three situations below happens:

(i) A path is found whose other endpoint (not v_0) is adjacent in G to some vertex y not in P. If y is an endpoint of some P_i then add the whole of P_i to P and go to some stage $\geq (r+3)$; otherwise just add y to P and go to stage $r+1$. (Observe that here green edges might be used.)

(ii) A path is found whose endpoints are adjacent in G. Either we found a Hamiltonian cycle or by connectivity in G' we can find a longer path without breaking the P_i's contained in P and thus go to stage $r+1$.

(iii) We run out of legal rotations to do. In this case proceed to Stage r part 2.

Stage r **part 2.** Now we have a (large) number of paths $Q_1, Q_2, \ldots$, each of length r, and each with an endpoint v_0 and (after some deletions) a different endpoint v_i. For $i = 1, 2, \ldots$ take Q_i, fix v_i, and run the procedure described in Stage r part 1, but without further recursion. Thus if we reach case (iii) above for all the Q_i then the algorithm fails.

The running time of the algorithm is dominated by Phase 4. (Phase 1 is easy since a.s. there are no small cycles with a common vertex.) In Part 1 of a stage we do $O(n)$ rotations, and each rotations can be carried out in $O(\log n)$ time (see Angluin and Valiant [1]); Part 2 requires $O(n)$ times as much work as Part 1; There are at most n stages. This justifies a time bound of $O(n^3 \log n)$.

3 Proof of the algorithm

In all what follows we assume that d is greater than a sufficiently large constant and at most $2 \ln n$. Other constants were chosen for convenience and no attempt was made to optimize them.

Throughout the proof we use the following bounds on the binomial distribution $B(n,p)$ without comment. For $0 \le \epsilon \le 1$

$$\Pr(B(n,p) \le (1-\epsilon)np) \le e^{\frac{1}{2}\epsilon^2 np};$$

$$\Pr(B(n,p) \ge (1+\epsilon)np) \le e^{\frac{1}{3}\epsilon^2 np}.$$

For $\alpha > 0$

$$\Pr(B(n,p) \ge \alpha np) \le \left(\frac{e}{\alpha}\right)^{\alpha np}.$$

Let $n_0 = ne^{-d/10}$ and $p = d/n$. The notation $[i]$ stands for the set $\{1, 2, \ldots, i\}$.

Lemma 1

$$\Pr(|X_0| \ge n_0) \le e^{-n_0}.$$

Proof: By definition X_0 is the set of vertices in G that have degree at most $d/4$ and therefore must have degree less than $(d/4 - 2)$ in $G \setminus H$. Therefore

$$\Pr(|X_0| \ge n_0) \le \binom{n}{n_0} \Pr([n_0] \subset X_0).$$

But if $[n_0] \subset X_0$ then any vertex in $[n_0]$ has at most $(d/4-2)$ neighbors in $[n_0+1, n]$. Hence

$$\begin{aligned}
\Pr(|X_0| \geq n_0) &\leq \binom{n}{n_0}\left(\sum_{0\leq i\leq d/4-2}\binom{n-n_0}{i}p^i(1-p)^{n-n_0-i}\right)^{n_0} \\
&\leq \binom{n}{n_0}e^{-n_0 d/4} \leq \left(\frac{ne^{1-d/4}}{n_0}\right)^{n_0} \\
&\leq e^{-n_0 d/10} \leq e^{-n_0}
\end{aligned}$$

□

Lemma 2 *If α is such that $1 < \alpha < d/10$, then any non-empty set of vertices $S \subset V$, of size*

$$s = |S| \leq \frac{n}{2e}\left(\frac{2\alpha}{ed}\right)^{\frac{\alpha}{\alpha-1}}$$

a.s. spans no more than αs edges in G.

Proof: Suppose a set S of s vertices spans at least αs edges. The edges in $S \cap H$ form a set of t disjoint paths of lengths $s_1, s_2, \ldots, s_t \geq 0$, where $1 \leq t \leq s$, and $s_1 + s_2 + \cdots + s_t = s - t$. The number of choices for these paths is at most $\binom{n}{t}\binom{s-1}{t-1}$. There are also at least $(\alpha-1)s+t$ random edges in S. So if we let $\beta = \alpha - 1$ then

$$\Pr(\exists S) \leq \sum_{1\leq t\leq s}\binom{n}{t}\binom{s-1}{t-1}\binom{\binom{s}{2}}{\beta s+t}\left(\frac{d}{n}\right)^{\beta s+t} \leq \sum_{1\leq t\leq s} u_t,$$

where

$$u_t = \binom{n}{t}2^s\binom{\binom{s}{2}}{\beta s+t}\left(\frac{d}{n}\right)^{\beta s+t}.$$

Now

$$\sum_{2\leq s\leq 10d}\sum_{1\leq t\leq s} u_t = O(n^{-\beta}).$$

We can therefore assume that $s > 10d$. In this case

$$\frac{u_{t+1}}{u_t} = \frac{n-t}{t+1}\cdot\frac{\binom{s}{2}-\beta s-t}{\beta s+t+1}\cdot\frac{d}{n} > 2.$$

Hence

$$\sum_{1\le t\le s} u_t \le 2u_s,$$

and

$$\begin{aligned}
\Pr(\exists S) &\le 2\binom{n}{s}2^s\binom{\binom{s}{2}}{\alpha s}\left(\frac{d}{n}\right)^{\alpha s} \\
&\le 2\left(\frac{ne}{s}\cdot 2\cdot\left(\frac{s^2ed}{2\alpha sn}\right)^{\alpha}\right)^s \qquad (2)\\
&= 2\left(\left(\frac{s}{n}\right)^{\alpha-1}\frac{2e^{\alpha+1}d^{\alpha}}{2^{\alpha}\alpha^{\alpha}}\right)^s
\end{aligned}$$

Finally, let

$$s_0 = \frac{n}{2e}\left(\frac{2\alpha}{ed}\right)^{\frac{\alpha}{\alpha-1}},$$

and observe that

$$\sum_{10d\le s\le s_0}\left(\left(\frac{s}{n}\right)^{\alpha-1}\frac{2e^{\alpha+1}d^{\alpha}}{(2\alpha)^{\alpha}}\right)^s = o(1).$$

$\square$

Lemma 3

$$\Pr(|X|\ge 5n_0)\le 2e^{-n_0}.$$

Proof: Define X_i inductively by

$$X_i = X_{i-1}\cup e_1\cup f_1\cup\ldots\cup e_K\cup f_K$$

(here we see each edge as a 2-element subset) and let $E_i = \{\text{edges } e\subseteq X_i\}$. Then

$$\frac{2K}{K+1}(|X_i|-|X_0|)\le |E_i|$$

is easily proved by induction on i. Hence

$$\frac{|E_i|}{|X_i|}\ge\frac{2K}{K+1}\left(1-\frac{|X_0|}{|X_i|}\right)$$

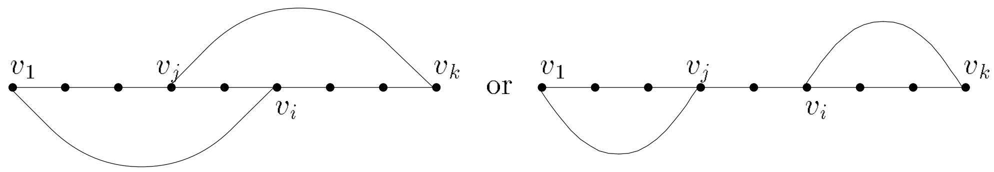

Figure 3: Linked cycle pairs.

Suppose that $|X| \geq 5n_0$. Now $|X_0| \leq n_0$ with probability at least $1 - e^{-n_0}$ and so we may assume that at some point $5n_0 \leq |X_i| \leq 5n_0 + K$. But then

$$\frac{|E_i|}{|X_i|} \geq \frac{2K}{K+1} \cdot \frac{4}{5} \geq \frac{16}{15}$$

if $K \geq 2$. Applying (2), the probability of this is at most e^{-n_0} and the lemma follows. □

Definition A *linked cycle pair* consists of two cycles joined together. (See figure 3) Such a pair can be viewed as a path $v_1, \ldots, v_k$ plus two extra edges, $\{v_1, v_i\}$ and $\{v_j, v_k\}$ with $1 \leq i, j \leq k$.

Lemma 4 *The graph G a.s. does not contain any linked cycle pair S such that $|X_0 \cap S| \geq |S|/10$.*

Proof: Lemma 1 implies that we can assume $|S| \leq 10n_0$.

Let $s = |S|$. We view S as a path P plus two extra edges, being made out of a set of t subpaths of H with lengths $s_1, s_2, \ldots s_t \geq 0$ plus $t+1$ edges from $G \setminus H$. Therefore

$$\mathbf{E}(\#\text{ linked cycle pairs}) \leq \sum_{t=1}^{s} \binom{n}{t} \binom{s-1}{t-1} s^2 2^{t+2} (t+2)! \left(\frac{d}{n}\right)^{t+1} \sum_{\substack{A \subseteq S \\ |A| > s/10}} \pi(A).$$

where $\pi(A)$ is the probability that some fixed set A is contained in X_0, that is that all vertices in some fixed set A have at most $d/4 - 2$ neighbors in $G \setminus H$.

Explanation of the inequality: consecutive terms estimate choices for starts of sub-paths of H; choices for $s_1, s_2, \ldots, s_t$; choices for the extra edges u_1 and u_2 (if a u edge is in H, then the corresponding subpath of H splits into two subpaths); direction and ordering of the subpaths along P; probability of the $t+1$ random edges; probability that the vertices in some A are in X_0. Observe that not all choices result in legal configurations.

Clearly $\pi(A) \leq e^{-d|A|/10}$. Hence

$$\begin{aligned}\mathbf{E}(\#\text{ linked cycle pairs}) &\leq \sum_{t=1}^{s} \frac{n^t}{t!} 2^s s^2 2^{t+2} (t+2)! \left(\frac{d}{n}\right)^{t+1} 2^s e^{-sd/100} \\ &\leq \frac{1}{n} \sum_{t=1}^{s} 8^s (s+2)^4 d^{t+1} e^{-sd/100} \qquad (3) \\ &\leq \frac{1}{n} 8^{s+1} (s+2)^4 d^{s+1} e^{-sd/100}\end{aligned}$$

Note that dist

$$\frac{1}{n} \sum_{s=1}^{\infty} 8^{s+1} (s+2)^4 d^{s+1} e^{-sd/100} = o(1).$$

for sufficiently large d. $\square$

Remark 1. The estimate in (3) shows that if $s \leq \frac{1}{2} \log_4 n$ then there are a.s. no linked cycle pairs, regardless of how many vertices of X_0 they might contain. In particular, there are a.s. no two triangles with a common vertex.

Lemma 5 *The graph G a.s. does not contain any linked cycle pair S such that $|X \cap S| \geq |S|/5$.*

Proof: Let $s = |S|$. We can assume by Remark 1 and Lemma 3 that

$$1000 \leq s \leq 25 n_0.$$

Suppose that G contains such a linked cycle pair and that s is as small as possible. Let S consist of a path P plus two extra edges e_1, e_2 as in the previous lemma. We start by proving the following

Claim 1 *For any vertex $v \in V$ there are at most 20 distinct paths of length 1 or 2 which begin at v and end in $S \cap X$.*

Proof: (of claim) Suppose that v is the endpoint of 20 paths of length at most 2 to $S \cap X$. (The number 20 was chosen for convenience.) Let the endpoints in $S \cap X$ be $x_1, x_2, \ldots x_{20}$ where x_i precedes x_{i+1} on P for $1 \leq i < 20$. Let Q_i, for $1 \leq i < 20$ denote the subpath of P joining x_i to x_{i+1}. It follows from Remark 1 that at most one of these paths has length less than 100. At most 6 contain an endpoint of e_1 or e_2. Also if $v \in S$ at most 2 of the x_i can be neighbors of v on P. We can therefore find t such that both of the paths Q_t, Q_{t+1} are of length at least 100 and neither contains an endpoint of e_1 or e_2.

Assume now that in fact $t = 1$. Let δ_i, for $i = 1, 2, 3$ be the number of internal vertices of the path P_i from v to x_i. (Thus $\delta_i = 0$ or 1.) Let $\delta = 1$ if $v \notin S$ and δ=0 if $v \in S$. Let Q_i contain k_i internal vertices out of which $k_{i,X}$ belong to X. Let k_X be the number of vertices in $S \cup X$.

Consider the set of vertices S_1 in $P_1 \cup P_2 \cup P_3 \cup Q_1 \cup Q_2$. Now $|S_1| < s$ and S_1 contains a linked cycle pair. Also S_1 has at most $k_1 + k_2 + \delta_1 + \delta_2 + \delta_3 + 4$ vertices, out of which at least $k_{1,X} + k_{2,X} + 3$ belong to X. Since S is minimal,

$$\frac{k_1 + k_2 + \delta_1 + \delta_2 + \delta_3 + 4}{k_{1,X} + k_{2,X} + 3} \geq 5. \tag{4}$$

Now consider the set of vertices S_2 in $(S \setminus (Q_1 \cup Q_2)) \cup \{x_1, x_3\} \cup P_1 \cup P_2$. The set S_2 also contains a linked cycle pair and $|S_2| < s$. Furthermore S_2 has at most $s - (k_1 + k_2 + 1) + \delta_1 + \delta_3 + \delta$ vertices out of which at least $k_X - (k_{1,X} + k_{2,X} + 1)$ belong to X. Hence

$$\frac{s - (k_1 + k_2 + 1) + \delta_1 + \delta_3 + \delta}{k_X - (k_{1,X} + k_{2,X} + 1)} \geq 5. \tag{5}$$

But (4) and (5) imply that

$$\begin{aligned} 5k_X &\leq s + 2\delta_1 + \delta_2 + 2\delta_3 + \delta - 7 \\ &\leq s - 1 \end{aligned}$$

This, of course, contradicts $k_X \geq s/5$, and concludes the proof of the claim. $\square$

Continuing with the proof of the Lemma, now let $S_1 = (X \setminus X_0) \cap S$. It follows from Lemma 4 that we may assume $|S_1| \geq s/10$.

Next let

$$N_i = \{x \in X \setminus S : {}_G(x, S_1) = i\} \qquad i = 1, 2.$$

Also let

$$\begin{aligned} S_{1,1} &= \{x \in S_1 : \mathrm{dist}_G(x, N_1) = 1\} \\ S_{1,2} &= \{x \in S_1 \setminus S_{1,1} : \mathrm{dist}_G(x, N_2) = 2\} \end{aligned}$$

Observe that because K is greater than 20, Claim 1 implies that $S_1 = S_{1,1} \cup S_{1,2}$, and therefore at least one of $|S_{1,1}|$ or $|S_{1,2}|$ is no less than $s/20$.

Case 1: $|S_{1,1}| \geq \frac{s}{20}$

Let $X' = X(G \setminus S, K - 20)$. We argue now that

$$N_1 \subseteq X \cap \bar{S} \subseteq X' \tag{6}$$

Consider the construction of $X(G, K) = X_0, x_1, x_2, \ldots, x_r$ and the construction of X'. Observe first that $X' \supseteq X_0(G \setminus S)$. We can assume that when constructing X' we always try first to add the lowest indexed $x_i \notin S$. We always succeed here, for if we have added $\{x_1, x_2, \ldots, x_{i-1}\} \cap \bar{S}$ and $e_1, \ldots, e_K, f_1, \ldots, f_K$ are the edges associated with the addition of x_i to $X(G, K)$ then removing S eliminates at most 20 pairs e_t, f_t (Claim 1). This completes the proof of (6)

Now it follows from Claim 1 that we can find a subset $S'_{1,1} \subseteq S_{1,1}, |S'_{1,1}| \geq \frac{1}{20}|S'_{1,1}| \geq \frac{s}{400}$ and a subset $N'_1 \subseteq N_1, |N'_1| = |S'_{1,1}|$ and a bijection $\phi : S'_{1,1} \to N'_1$ such that G contains the edges $\{(v, \phi(v))\}$.

We can now proceed as in the proof of Lemma 4 except that we replace $\pi(A)$ by $\pi'(A)$, which is the probability that exists $B \subseteq X'$ such that $|B| = |A|$, and there exists a bijection $\phi : A \to B$ such that $\forall v \in A, (v, \phi(v)) \in G$

But, where $a = |A|$,

$$\begin{aligned} \pi'(A) &\leq \sum_{b=0}^{a} \binom{a}{b} 2^b \binom{n}{a-b} (a-b)! \left(\frac{d}{n}\right)^{a-b} e^{-ad/15} \\ &\leq a 4^a d^a e^{-ad/15} \leq e^{-ad/20} \end{aligned}$$

Here b refers to the number of edges $(v, \phi(v))$ which are in H and the term $e^{-ad/15}$ arises as follows: having fixed S, B, the edges inside S, and the edges between S and B, we can argue that, by symmetry, X' is a random $|X'|-$subset of $\bar{S}$ and so

$$\begin{aligned}
\Pr(B \subseteq X') &= \mathbf{E}_{|X'|}\left(\binom{|X'|}{a} \Big/ \binom{n-s}{a}\right) \\
&\leq \Pr(B \subseteq X' \mid |X'| \leq 5(n-s)e^{-d/10}) \\
&\qquad + \Pr(|X'| \geq 5(n-s)e^{-d/10}) \\
&\leq \mathbf{E}_{|X'|}\left(\binom{|X'|}{a} \Big/ \binom{n-s}{a} \mid |X'| \leq 5(n-s)e^{-d/10}\right) \\
&\qquad + 2e^{-(n-s)e^{-d/10}} \\
&\leq (5e^{-d/10})^a + 2e^{-(n-s)e^{-d/10}} \leq e^{-ad/15}
\end{aligned}$$

The rest of the proof for this case is as in Lemma 4.

Case 2: $|S_{1,2}| \geq \frac{s}{20}$

It follows from Claim 1 that we can find a subset $S_{1,2} \subseteq S_{1,2}$ with $|S'_{1,2}| \geq |S_{1,2}|/40 \geq s/800$ and $N'_2 \subseteq N_2$ such that there is a set of $|N'_2| = |S'_{1,2}|$ vertex disjoint paths of length 2 from N'_2 to $S'_{1,2}$. Since $K > 24$ we can furthermore assume that none of the edges of these paths are part of the cycles C_1, C_2 even though the internal vertices may actually be in S.

We can then proceed as in Case 1 with $\pi'(A)$ replaced by $\pi''(A)$ defined as the probability that there exists $B_1, B_2 \subseteq V$ with $|B_1| = |B_2| = |A|$, such that $B_2 \subseteq X(G \setminus (S \cup B_1), K-20)$, and there exists bijections $\phi_1 : B_1 \to A$ and $\phi_2 : B_2 \to B_1$ such that $\forall v_i \in B_i, i = 1,2) : (v_1, \phi_1(v_1)), (v_2, \phi_2(v_2)) \in G$.

Letting b_i run over the number of edges $(v_i, \phi(v_i))$ which are in H for $i = 1, 2$ we find that

$$\begin{aligned}
\pi''(A) &\leq \sum_{b_1=0}^{a} \binom{a}{b_1} 2^{b_1} \binom{n}{a-b_1} (a-b_1)! \left(\frac{d}{n}\right)^{a-b_1} \\
&\qquad \times \sum_{b_2=0}^{a} \binom{a}{b_2} 2^{b_2} \binom{n}{a-b_2} (a-b_2)! \left(\frac{d}{n}\right)^{a-b_2} e^{-ad/15} \\
&\leq a^2 16^a d^{2a} e^{-ad/15} \leq e^{-ad/20}
\end{aligned}$$

and this completes the proof of the lemma. □

A *path cover of* X is a set of vertex disjoint paths containing X all of whose endpoints are outside X. Lemma 5 implies that a.s. each component of the graph G_X defined at the beginning of Phase 2 is either a tree or a unicyclic graph. (If not we have a linked cycle pair S with $S \cap X \geq |S|/5$.) For such graphs there are fast dynamic programming algorithms for finding a path cover of the interior points, if one exists – linear for trees, at worst quadratic for unicyclic components. The argument here is even simpler than in [9], Lemma 3.2. On the other hand G_X contains a path cover of X because G is Hamiltonian.

Lemma 6 *Any set $S \subset V \setminus Y$, whose size satisfies*

$$1 \leq |S| \leq \frac{n}{50e^3(K+3)^3},$$

almost surely has

$$|N_b(S)| > (K+2)\,|S|,$$

where $N_b(S) = \{w \notin S \mid \exists v \in S, \{v,w\} \in E_b(G)\}$.

Proof: Suppose that there exists $S \subseteq V \setminus Y$ with $|N_b(S)| \leq (K+2)|S|$. Then if $T = S \cup N_b(S)$ we have

$$|T| \leq (K+3)|S|$$

and T spans at least $\frac{d}{10}|S|$ edges in G_b and hence in G as well.

Assuming d is large and applying Lemma 2 with $\alpha = \frac{d}{10K+30}$ we see that we must have

$$\begin{aligned} |T| &> \frac{n}{2e}\left(\frac{d}{10K+30}\frac{2}{ed}\right)^{d/(d-10K-30)} \\ &> \frac{n}{2e}\left(\frac{1}{5e(K+3)}\right)^{2} \end{aligned}$$

and the result follows. $\square$

Remark 2: Suppose $v \notin Y$ (recall that Y is the set of vertices internal to the paths in $\mathcal{P}$.) Then we claim that v has at most K neighbors in Y, since any v that has $K+1$ neighbors in Y then v is in X by construction. (We need to use the fact that no two triangles share a common vertex, a.s. – see Remark 1.)

Lemma 7 *The graph G' obtained at the end of Phase 2 of the algorithm, is almost surely connected.*

Proof: Suppose that G' has a component A of size $\leq n/2$. For each path $P \in \mathcal{P}$ we see that A contains all of the vertices of P or none of them. Now consider $A' = A \setminus Y$. If $|A'| \leq n_1 = n/(50e^3(K+3)^3)$ then from Lemma 6, A' has at least $(K+2)|A'|$ neighbors outside $|A'|$. But at most $(K-1)|A'|$ of these can be in $A \cap Y$ (Remark 2.) This deals with $|A| \leq n_1$. For $|A| \geq n_1$ observe that if $S : \bar{S}$ denotes the set of H-edges with one end in S then

$$\begin{aligned}
&\Pr\left(\exists A \text{ s.t. } |A| = a,\ n_1 \leq a \leq \tfrac{1}{2}n, \text{ and } |A : \bar{A}| \leq a(n-a)d/(2n)\right) \\
&\qquad \leq \sum_{n_1 \leq a \leq n/2} \binom{n}{a} \exp\left(-\frac{1}{8}\frac{a(n-a)d}{n}\right) \\
&\qquad \leq n2^n \exp\left(-\frac{dn}{500e^3(K+3)^3}\right) \\
&\qquad = o(1)
\end{aligned}$$

Thus we can a.s. assume that $|A : \bar{A}| \geq n_1(n-n_1)d/(2n)$.

On the other hand we are unlikely to have deleted this many edges in going from G to G'. Observe

$$\begin{aligned}
&\Pr\left(\exists S \text{ s.t. } |S| \leq 5n_0 \text{ and } |S : \bar{S}| \geq n_2 = \tfrac{1}{2}n_1(n-n_1)d/(2n)\right) \\
&\qquad \leq \sum_{1 \leq s \leq 5n_0} \binom{n}{s} \left(\frac{2es(n-s)}{n_1(n-n_1)}\right)^{n_2} = o(1).
\end{aligned}$$

and the lemma follows as we have only deleted edges of G incident with X and Lemma 2 implies that X a.s. meets less than $\frac{1}{2}n_2$ edges. $\Box$

To complete our analysis we need to estimate the failure probability in Phase 4. Assume that in a certain stage $r \leq n$ the algorithm failed, that is all legal rotations were exhausted and no cycle closing was possible. Let P be the path used in stage r part 1. Let the endpoints of P be v_0 and v_1 and let $\mathrm{END}(v_0)$ denote the endpoints found when rotating with v_0 fixed. Similarly, for each $v \in \mathrm{END}(v_0)$ let $\mathrm{END}(v)$ be the endpoints found when rotating with v fixed in stage r part 2. Clearly the failure of the algorithm means that

$$\begin{aligned} &\forall v \in \{v_0\} \cup \mathrm{END}(v_0) \\ &\qquad w \in \mathrm{END}(v) \Longrightarrow \{v, w\} \notin E(G); \end{aligned} \tag{7}$$

otherwise a cycle extension applies.

Let x, y, z be three consecutive vertices within the initial order of a path Q with endpoint v fixed. Pósa [14] observed that if $w \in \mathrm{END}(v)$ and $y \in N(w) \setminus \mathrm{END}(v)$, and all rotations are allowed, then one of x or z is also in $\mathrm{END}(v)$. In our case we need to take into account that only blue edges are used for rotations, and rotations that split the P_i subpaths are not allowed.

But since we took care to insure that a vertex in $\mathrm{END}(v)$, has at most K neighbors in Y in G, a fortiori it has at most K neighbors in Y in G_b. Hence

$$|N_b(\mathrm{END}(v)) \cap Y| \leq K\,|\mathrm{END}(v)|$$

Consider now the initial ordering of the path Q with endpoint v fixed, and consider all $w \in \mathrm{END}(v)$. Recall that $N_b(S)$ is the set of neighbors of S outside S. By Pósa's argument the number of vertices in $N_b(\mathrm{END}(v)) \setminus Y$ is at most $2\,|\mathrm{END}(v)|$.

Therefore

$$\begin{aligned} &\forall v \in \{v_0\} \cup \mathrm{END}(v_0) \\ &\qquad |N_b(\mathrm{END}(v))| \leq (K+2)\,|\mathrm{END}(v)| \end{aligned} \tag{8}$$

and so by Lemma 6

$$\begin{aligned} &\forall v \in \{v_0\} \cup \mathrm{END}(v_0) \\ &\qquad |\mathrm{END}(v)| > \frac{n}{50e^3(K+3)^3} \end{aligned} \tag{9}$$

We can see immediately that (7) requires ϵn^2 non-edges for some constant $\epsilon > 0$ (independent of d) and this event is unlikely even for the sparse random graphs that we consider. The precise way to make these estimations follows the techniques from [8], [9] and [10].

Observe that after converting G into a blue-green instance in Phase 3, the rest of the algorithm is deterministic and *actually* uses at most n green edges, because each use of a green edge results in an extension. Let U be the set of green edges used by the algorithm up to stage where it fails. Call a set $D \subset E_g$ of $\lceil \ln n \rceil$ green edges, *deletable* from E_g if it is disjoint from U. Clearly if D is deletable, the run of Phase 4 using $E_g \setminus D$ instead of E_g is identical to the run using E_g.

By conditioning we may assume that we work in a random graph model with a fixed number of blue and green edges (about $nd/4$ each). For the purpose of the failure probability estimation only, we delete a random set D of $l = \lceil \ln n \rceil$ green edges and consider the following two events:

$\mathcal{E}_1 =$ All the a.s. properties described in the lemmas hold and yet the algorithm fails.

$\mathcal{E}_2 =$ $\mathcal{E}_1$ and D is deletable.

We shall prove that as $n \to \infty$

$$\Pr(\mathcal{E}_2 \mid \mathcal{E}_1) \geq (1 - O(1/d))^l, \tag{10}$$

and

$$\Pr(\mathcal{E}_2) \leq (1 - \epsilon)^l, \tag{11}$$

for a certain constant ϵ.

We can then use

$$\Pr(\mathcal{E}_1) \leq \Pr(\mathcal{E}_2) / \Pr(\mathcal{E}_2 \mid \mathcal{E}_1).$$

The proof of (10) is easy since $|E_g|$ is about $nd/4$ and D has to avoid at most n edges of U. To prove (11) notice that if $\mathcal{E}_2$ holds then the algorithm fails on G and $G \setminus D$ in exactly the same way and at the same stage r. In other words, if we condition on $G \setminus D$, the event $\mathcal{E}_2$ holds only if the addition of l random green edges to the graph in stage r does not allow continuation to stage $r+1$. But that means that the added green edges do not fall into any of the ϵn^2 forbidden spots, which happens with probability at most $(1-\epsilon)^l$.

4 Open problems

A more careful analysis would probably yield a smaller lower bound on the minimum density required for the algorithm to work. However, the approach presented here is not likely to be extendable to the situation when the input is a Hamiltonian cycle plus a random perfect matching or to similar very small degree inputs. The former problem appears to be amenable to an attack based on recent results of Robinson and Wormald [15, 16], and Jerrum [13].

Acknowledgement

We would like to thank Joan Feigenbaum who greatly helped us understand and expose the connections between the question considered here and zero-knowledge protocols. We would also like to thank Tomasz Łuczak whose comments have helped to make the paper a little more readable.

References

[1] D. Angluin and L. G. Valiant. Fast probabilistic algorithms for Hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18:155–193, 1979.

[2] B. Bollobás. *Random Graphs.* Academic Press, 1985.

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

[4] A. Z. Broder, A. M. Frieze, and E. Shamir Finding hidden Hamiltonian cycles In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 182–189, May 1991.

[5] R. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, pages 280–285, October 1987.

[6] T. Bui, S. Chaudhuri, T. Leighton, and M. Sipser. Graph bisection algorithms with good average case behavior. *Combinatorica*, 6, 1986.

[7] M. E. Dyer and A. M. Frieze. Fast algorithms for some random np-hard problems. *Journal of Algorithms*, 10:451–489, 1989.

[8] T. I. Fenner and A. M. Frieze On the existence of Hamiltonian cycles in a class of random graphs. *Discrete mathematics*, 45:301–305, 1983.

[9] A. M. Frieze. On the exact solution of random traveling salesman problems with medium-sized integer costs. *SIAM Journal on Computing*, 16:1052–1072, 1987.

[10] A. M. Frieze. Finding Hamilton cycles in sparse random graphs. *Journal of Combinatorial Theory B*, 44:230–250, 1988.

[11] Y. Gurevich and S. Shelah. Expected computation time for Hamiltonian path problem. *SIAM Journal on Computing*, 16(3):486–502, 1987.

[12] L. Kučera and S. Micali. Cryptography and random graphs. Unpublished manuscript, 1988.

[13] M. R. Jerrum. Private communication.

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

[15] R. W. Robinson and N. C. Wormald. Almost all cubic graphs are Hamiltonian. *Random Structures and Algorithms* 3:117–126, 1992.

[16] R. W. Robinson and N. C. Wormald. Almost all regular graphs are Hamiltonian. *Random Structures and Algorithms*, to appear.

[17] E. Shamir. How many edges make a graph Hamiltonian? *Combinatorica*, 3:123–132, 1983.

[18] J. S. Turner. On the probable performance of heuristics for bandwidth minimization. *SIAM Journal on Computing*, 15(2):561–580, 1986.