Theoretical Computer Science 407 (2008) 545–553

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Testing whether a digraph contains H-free k-induced subgraphs[☆]

Hong-Yiu Lin[a], Yuh-Dauh Lyuu[a,*], Tak-Man Ma[b], Yen-Wu Ti[a]

[a] *Department of Computer Science and Information Engineering, National Taiwan University, No.1, Sec. 4, Roosevelt Road, Taipei 106, Taiwan*

[b] *MediaTek Inc., Hsinchu, Taiwan*

ARTICLE INFO

Article history:
Received 29 April 2007
Received in revised form 9 June 2008
Accepted 25 August 2008
Communicated by P. Spirakis

Keywords:
Directed graphs
Property testing
Randomized algorithm

ABSTRACT

A subgraph induced by k vertices is called a k-induced subgraph. We prove that determining if a digraph G contains H-free k-induced subgraphs is $\Omega(N^2)$-evasive. Then we construct an ϵ-tester to test this property. (An ϵ-tester for a property Π is guaranteed to distinguish, with probability at least $2/3$, between the case of G satisfying Π and the case of G being ϵ-far from satisfying Π.) The query complexity of the ϵ-tester is independent of the size of the input digraph. An (ϵ, δ)-tester for a property Π is an ϵ-tester for Π that is furthermore guaranteed to accept with probability at least $2/3$ any input that is δ-close to satisfying Π. This paper presents an (ϵ, δ)-tester for whether a digraph contains H-free k-induced subgraphs.

1. Introduction

A classical computational problem is to verify if an object has a predetermined property such as the connectedness of a graph or the transitivity of a digraph. Unfortunately, sometimes no fast algorithms exist that give the exact answer. In these cases, an approximate answer within a reasonable complexity is an attractive alternative.

A *property-testing algorithm* offers such answers. It determines whether a problem instance has a certain property or is far from any instances having such property. It is, however, arbitrary on problem instances falling between the two categories. The general notion of property testing was first formulated by Rubinfeld and Sudan [1].

A testing algorithm of property Π can make queries on the incidence relations of vertices in an input graph G. Property Π is *$\Omega(N^2)$-evasive* if there is no deterministic testing algorithm with query complexity $o(N^2)$ that can correctly decide if the input has Π. Holt and Reingold [2] were the first to consider the complexity of recognizing graph properties from their adjacency matrix representations. They showed that the graph properties of connectivity and the existence of cycles are both $\Omega(N^2)$-evasive. An important open problem in this area is the Aanderaa–Rosenberg conjecture [2–5]: Every nontrivial monotone graph property without self-loops is $\binom{N}{2}$-evasive. Rivest and Vuillemin [6] resolved a weaker version of the Aandreaa–Rosenberg conjecture. The weaker version says that every nontrivial monotone graph property has decision tree complexity $\Omega(N^2)$.

An *ϵ-tester* (or simply a *tester*) for a digraph property Π is a randomized algorithm that is given a size parameter N, a distance parameter ϵ and the ability to make queries as to whether a directed edge of the input digraph G with N vertices exists. The total number of queries is called the *query complexity* of the tester. Let $\{g_i\}$ be the set of digraphs with N vertices that satisfy Π. The algorithm needs to distinguish with probability at least $2/3$ between the case of G having Π and the case

☆ This research was supported in part by the National Science Council of Taiwan under NSC grant 97-2221-E-002-096-MY3 and Excellent Research Projects of National Taiwan University under grant 97R0062-05.

* Corresponding author. Tel.: +886 2 33664888; fax: +886 2 23628167.
E-mail addresses: r93028@csie.ntu.edu.tw (H.-Y. Lin), lyuu@csie.ntu.edu.tw (Y.-D. Lyuu), mata3@seas.upenn.edu (T.-M. Ma), d91010@csie.ntu.edu.tw (Y.-W. Ti).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.08.033

of G differing from any g_i in at least $\epsilon\binom{N}{2}$ edges [7]. In the latter case, G is said to be *ϵ-far* from property Π. The probability 2/3 can be replaced by any constant smaller than 1 because the algorithm can be repeated if necessary.

A graph property is *testable* if the property has a tester and the total number of queries is $o(N^2)$. The testing of graph properties was pioneered by Goldreich, Goldwasser and Ron [8]. They showed that all graph properties describable by the existence of a partition of a certain type are testable. For a fixed digraph H with at least one edge, let P_H denote the property of the input digraph being *H-free*. In other words, the digraph G has P_H if and only if it contains no subgraphs isomorphic to H. Alon and Shapira [7] proved that P_H is testable with the total number of queries bounded only by a function of ϵ, independent of N. This result has been improved later by Alon and Shapira [9]. Alon, Fischer, Krivelevich and Szegedy [10] showed that every first-order undirected graph property without a quantifier alternation of type "$\forall\ \exists$" has ϵ-testers whose query complexity is independent of the size of the input digraph. More recently, Alon, Fischer, Newman and Shapira [11] proved a very general result for undirected graphs, which says that the property defined by having any given Szemerédi-partition is testable with a constant number of queries. Moreover, a purely combinatorial characterization of the graph properties is testable with a constant number of queries. The testing of other graph and combinatorial properties has also been intensively studied [12–16].

An input is *δ-close* to having property Π if it is not δ-far from having property Π. Recall that an ϵ-tester says nothing about inputs which are δ-close to having property Π. An (ϵ, δ)-*tester* is an ϵ-tester that is furthermore guaranteed to accept with probability at least 2/3 any input that is δ-close to having a property Π. This type of property tester was first studied by Parnas, Ron and Rubinfeld [17]. Fischer and Newman [18,19] proved that a testable property is also (ϵ, δ)-testable when $\delta < \epsilon$.

A subgraph induced by k vertices is called a *k-induced subgraph*, and a subgraph with k vertices is called a *k-subgraph*. Recall that a k-induced subgraph includes *all* the existing edges between the said k vertices. This paper studies property testing for the existence of H-free k-induced subgraphs for digraphs, given a digraph H. Usually, a problem on digraphs is more difficult than the otherwise identical problem on undirected graphs. We say a digraph G has property $P_{k,H}$ if and only if it contains an H-free k-induced subgraph. This paper proves that property $P_{k,H}$ is $\Omega(N^2)$-evasive. (Since $P_{k,H}$ is not a monotone graph property, we can not use Rivest and Vuillemin's result [6] to prove it is $\Omega(N^2)$-evasive.) For any digraph H Alon and Shapira defined a binary function called $f(\epsilon; H)$. We prove that for every digraph H whose $f(\epsilon; H)$ is not too small, there exist an ϵ-tester and an (ϵ, δ)-tester for $P_{k,H}$. For a graph A, core denotes a graph B such that there exists a homomorphism from A to B and B is minimal in the number of vertices with this property. Alon and Shapira [7] proved that for a connected H, $f(\epsilon; H)$ has a polynomial dependency on $1/\epsilon$ if and only if the core of H is either an oriented tree or a directed cycle of length 2. In this case, we can find a smaller ϵ such that $f(\epsilon; H)$ satisfies the restrictions of our testing algorithms. In both results, the query complexity is dependent on ϵ but independent of the input size.

We use induced subgraphs instead of merely subgraphs (as in the property of H-freeness, P_H) in the definition of $P_{k,H}$. This is because, assuming H is not a digraph with no edges, then every digraph trivially contains H-free k-subgraphs: A subgraph with k isolated vertices is an H-free k-subgraph. On the other hand, some ϵ-far instances for the previously studied H-freeness property may become acceptable instances of $P_{k,H}$. For example, assume $G_1 = (V_1, E_1)$ is a digraph with $N - k$ vertices that contains a subgraph isomorphic to H and $G_2 = (V_2, E_2)$ is a digraph with k isolated vertices. If we combine G_1 and G_2 into $G_3 = (V_1 \cup V_2, E_1 \cup E_2)$, then G_3 is not an H-free digraph but is a digraph with an H-free k-induced subgraph. The existence of H-free subgraphs has been investigated by some recent study [21], so it will be interesting to study property testing for $P_{k,H}$.

Our paper is organized as follows. In Section 2 we use Turán numbers to prove that testing the existence of H-free k-induced subgraphs is $\Omega(N^2)$-evasive. In Section 3 we prove that $P_{k,H}$ is testable. In Section 4 we use a technique similar to that used in Section 3 to prove $P_{k,H}$ is (ϵ, δ)-testable. In Section 5 we conclude.

2. Existence of H-free k-induced subgraphs Is $\Omega(N^2)$-evasive

In this section, we show that the query complexity of any deterministic algorithm for the existence of H-free k-induced subgraphs is $\Omega(N^2)$.

First, we need some results concerning Turán numbers. For any integer N and a fixed graph H, let $\mathrm{ex}(N, H)$ denote the maximum number of edges that an N-vertex graph may have if it contains no isomorphic copy of H. This is the *Turán number* of H. Furthermore, we will denote by $b_{r,s}$ the complete undirected bipartite graph between a set of r vertices and another set of s vertices. The following fact is well-known.

Fact 1 (*[20]*). *For* $r \le s$, $\mathrm{ex}(N, b_{r,s}) = O(N^{2-(1/r)})$.

If we replace the undirected edges of $b_{r,s}$ by directed edges with an arbitrary direction, a complete bipartite digraph $d_{r,s}$ results. The next theorem shows that it is $\Omega(N^2)$-evasive to determine if there is a $d_{r,s}$-free k-induced subgraph in a digraph. In our model, whenever an algorithm queries a pair of vertices x, y in the input graph, it actually means that the algorithm queries the existence of edges (x, y) and (y, x) simultaneously. For a set S, we say that a subset $T \subseteq S$ is a k-subset of S if $|T| = k$. If a digraph G contains a subgraph isomorphic to a digraph H, then we say that G contains a copy of H.

Theorem 2. *For any constant $\rho < 1$, $k < N/2$ and any complete bipartite digraph $d_{r,s}$, no algorithm can determine whether a digraph contains a $d_{r,s}$-free k-induced subgraph with query complexity $\le \rho\binom{k}{2}$ if $k = \lambda N$ with λ being a constant.*

Proof. Suppose there exists an algorithm A that determines if a digraph contains a $d_{r,s}$-free k-induced subgraph with $\rho\binom{k}{2}$ queries. For the rest of the proof, assume $k = O(N)$ and k is large enough so that

$$(1-\rho)\binom{k}{2} \geq \mathrm{ex}(k, b_{r,s}) = O(k^{2-(1/r)}). \tag{1}$$

Start with a digraph G_1 with N vertices that contains no copies of $d_{r,s}$ (this is easy to construct). Let G_1 be the input of A. Obviously, all k-induced subgraphs of G_1 are $d_{r,s}$-free. Let $G_2 = (V_2, E_2)$ be a graph with N isolated vertices. Every time A queries a pair of vertices x, y in G_1, we add that edge to G_2 if there is an edge between them. When A stops, the resulting G_2 has no k-induced subgraphs which contain $d_{r,s}$, just like G_1. For those vertex pairs of G_1 that are not queried by A, we add an edge (but without the directions) to G_2. For each k-induced subgraph of G_2, at least $(1-\rho)\binom{k}{2}$ undirected edges are added. According to Fact 1, every k-induced subgraph of G_2 must contain a copy of $b_{r,s}$ with the undirected edges alone because of Eq. (1).

Now, we select a k-induced subgraph K_1 in G_2 and replace one copy of $b_{r,s}$ in K_1 by $d_{r,s}$. Let $V_{b,1}$ be the vertex set of this copy of $d_{r,s}$, and define $h = |V_{b,1}| = r + s$. For each subset of V_2 with size k that contains $V_{b,1}$, its induced subgraph has a copy of $d_{r,s}$ too. There are $\binom{N-h}{k-h}$ such k-subsets of V that contain $V_{b,1}$. Let $k/N = \lambda$. Recall that λ is a constant. Now, the ratio of the number of all such k-subsets to the number of k-induced subgraphs of G_2 is $\binom{N-h}{k-h}/\binom{N}{k}$. Note that

$$\frac{\binom{N-h}{k-h}}{\binom{N}{k}} = \frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)}.$$

As h is a constant and $k < N/2$, it is not hard to prove that there is a number $m > 0$ such that for every $N > m$ it holds that

$$\frac{k}{N} > \frac{k-1}{N-1} > \cdots > \frac{k-h+2}{N-h+2} > \frac{k-h+1}{N-h+1} = \frac{(k/N)-(h/N)+1/N}{1-(h-1)/N} > \frac{\lambda}{1+\lambda}.$$

Thus if N is large enough,

$$\frac{\binom{N-h}{k-h}}{\binom{N}{k}} = \frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)} > \left(\frac{\lambda}{1+\lambda}\right)^h.$$

We conclude that at least $\left(\frac{\lambda}{1+\lambda}\right)^h\binom{N}{k}$ k-induced subgraphs contain a copy of $d_{r,s}$.

Next we select another k-induced subgraph $K_2 = (V', E')$ with $V' \cap V_{b,1} = \emptyset$. It is worth noting that K_2 also has a copy of $b_{r,s}$, and the vertex set of $b_{r,s}$ is $V_{b,2}$. Like what we did before, we replace this copy of $b_{r,s}$ in K_2 by $d_{r,s}$. There are $\binom{N-2h}{k-h}$ such k-subsets of V that contain V_2. The ratio of the number of all such k-subsets to the number of k-induced subgraphs of G_2 is $\binom{N-2h}{k-h}/\binom{N}{k}$. Again, for N large enough,

$$\lim_{N\to\infty}\frac{\binom{N-2h}{k-h}}{\binom{N}{k}} = \lim_{N\to\infty}\frac{\binom{N-h}{k-h}}{\binom{N}{k}} > \left(\frac{\lambda}{1+\lambda}\right)^h.$$

We claim that in general, for every constant i,

$$\begin{aligned}\frac{\binom{N-ih}{k-h}}{\binom{N}{k}} &= \frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)}\cdot\frac{(N-k)\cdots(N-k-(i-1)h+1)}{(N-h)\cdots(N-ih+1)}\\ &> \left(\frac{\lambda}{1+\lambda}\right)^h.\end{aligned} \tag{2}$$

To verify this, recall that as we showed before,

$$\frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)} > \left(\frac{\lambda}{1+\lambda}\right)^h.$$

As for

$$\frac{(N-k)\cdots(N-k-(i-1)h+1)}{(N-h)\cdots(N-ih+1)},$$

since

$$\frac{N-k}{N-h} > \frac{N-k-1}{N-h-1} > \cdots > \frac{N-k-(i-1)h+1}{N-ih+1}$$

we have

$$\frac{(N-k)(N-k-1)\cdots(N-k-(i-1)h+1)}{(N-h)(N-h-1)\cdots(N-ih+1)} > \left(\frac{N-k-(i-1)h+1}{N-ih+1}\right)^{(i-1)h}.$$

Now, with $k = \lambda N$, it is easy to see that

$$\frac{N-k-(i-1)h+1}{N-ih+1} > \frac{N-k-(i-1)h+1}{N} > (1-2\lambda)$$

where the last inequality is due to $k > (i-1)h - 1$. Hence, when we repeat the above process i times, at least

$$[(1-2\lambda) + (1-2\lambda)^2 + \cdots + (1-2\lambda)^{(i-1)h}]\left(\frac{\lambda}{1+\lambda}\right)^h \binom{N}{k} \tag{3}$$

k-induced subgraphs contain a copy of $d_{r,s}$. Recall that $k < N/2$. Hence $2\lambda < 1$ and formula (3) is less than $\frac{1}{2\lambda}\left(\frac{\lambda}{1+\lambda}\right)^h \binom{N}{k}$.

Since λ, h and $(\lambda/(1+\lambda))^{-h}$ are constants, we can repeat this process $2\lambda(\lambda/(1+\lambda))^{-h}$ times such that $V_{b,i} \cap V_{b,j} = \emptyset$ for $i \neq j$ and N large enough. After having repeated this process that many times, we select $2\lambda(\lambda/(1+\lambda))^{-h}h < N$ distinct vertices from V for N large enough, and, by Eq. (2), the ratio of the number of $K_{2\lambda(\lambda/(1+\lambda))^{-h}}$ to the number of all k-induced subgraphs of G_2 will be at least $2\lambda(\lambda/(1+\lambda))^{-h}$. The number of k-induced subgraphs that contain a copy of $d_{r,s}$ then is at least $2\lambda(\lambda/(1+\lambda))^{-h}\frac{(\lambda/(1+\lambda))^h}{2\lambda}\binom{N}{k} = \binom{N}{k}$. In other words, after we repeat this process $2\lambda(\lambda/(1+\lambda))^{-h}$ times and remove the remaining undirected edges, all k-induced subgraphs of G_2 will have a copy of $d_{r,s}$. This digraph G_2 contains, therefore, no H-free k-induced subgraph. However, A cannot distinguish between G_1 and G_2 because we have only changed G_2's unqueried edges. So, A will accept G_2, which is a contradiction. ■

3. An ϵ-tester

Fix a digraph H with h vertices and $m \geq 1$ edges. Recall that $P_{k,H}$, where $k \geq h$, denotes the property that G contains an H-free k-induced subgraph. We will show that property $P_{k,H}$ is testable with a query complexity independent of the input size. A set with size n will be called an n-set, and a multiset with size n will be called an n-multiset. There is a function $f(\epsilon; H)$ with the following properties, which will be critical to our analysis later.

Theorem 3 ([7]). *Let H be a fixed digraph with h vertices and D be a digraph with N vertices. If at least ϵN^2 edges have to be removed from D to make it H-free, then D contains at least $f(\epsilon; H)N^h$ copies of H.*

The following corollary is immediate.

Corollary 4. *Let H be a fixed digraph with h vertices and m edges, D be a digraph with N vertices and $\sigma = \binom{\binom{h}{2}}{m}$. If at least ϵN^2 edges have to be removed from D to make it H-free, then D contains at least $f(\epsilon; H)N^h/\sigma$ h-sets whose induced subgraphs contain copies of H.*

Suppose the input N-vertex digraph $G = (V, E)$ is ϵ-far from having property $P_{k,H}$. Corollary 4 tells us that G must contain at least $f(\epsilon; H)N^h/\binom{\binom{h}{2}}{m}$ h-sets whose induced subgraphs contain copies of H. So to test property $P_{k,H}$ on G, our idea is to randomly select many h-sets from V. Suppose G contains an H-free k-induced subgraph, say (V_k, E_k). Then with enough h-sets from V, at least one of them is expected to be a subset of V_k with high probability. In these cases, we will check if an h-set S satisfies $S \subseteq V_k$ in 2 steps. First, we check the induced subgraph of S. When $S \subseteq V_k$, the induced subgraph of S contains no copies of H. If the induced subgraph of S contains no copies of H, we randomly add a number of other vertices to S (the number will be determined later) and check if there is a subset of S (with a size to be determined later) whose induced subgraph contains no copies of H. If $S \subseteq V_k$, we expect that S will pass these tests with high probability. Thus, G will be accepted by our algorithm with high probability. On the other hand, suppose G is ϵ-far from any digraph which has property $P_{k,H}$. Then we expect to find a copy of H in all the induced subgraphs of the above-mentioned h-sets S with high probability. Our algorithm is detailed in Fig. 1.

We shall need the Chernoff bound in later analysis.

Theorem 5 (*Chernoff Bound*). *Let $X = X_1 + X_2 + \cdots + X_n$ be a sum of n independent random variables such that $0 < \Pr[X_i = 1] < 1$ holds for each $i = 1, 2, \ldots, n$ and $\mu = E[X]$. Then for any $0 < \Delta < 1$,*

$$\Pr[X < (1-\Delta)\mu] < e^{-\mu\Delta^2/2}$$

where e *is the base of the natural logarithm.*

Note that in property $P_{k,H}$, h is a constant. Hence $f(\epsilon; H)$ is a function in ϵ only. We assume that H is a fixed digraph with h vertices and m edges and recall that G is the input digraph with N vertices from now on.

1: **if** $k < \sqrt{\epsilon}N$ **then**
2: **ACCEPT**
3: **end if**
4: let $\lambda = k/N$, $\kappa = \log_{1-\frac{(\sqrt{\epsilon})^h}{2}}(1/6)$, $\sigma = \binom{\binom{h}{2}}{m}$ and $\theta = \max\{\log_{\frac{6f(\epsilon;H)h!}{\sigma\lambda^2}}(2/3)^{1/\kappa}, 1\}$
5: **for** $i = 1$ to κ **do**
6: randomly select an h-set S from V
7: **if** the induced subgraph of S does not contain an H **then**
8: randomly select additional vertices $p = 6\theta h/\lambda$ times (with replacements) from $V - S$ (assume these p vertices to be $x_1, x_2, \ldots, x_p$) {note there are $\binom{p}{\theta h}$ (θh)-multisets in $\{x_1, x_2, \ldots, x_p\}$}
9: **for** $j = 1$ to $\binom{p}{\theta h}$ **do**
10: let S_j be the jth (θh)-multiset selected in step 8
11: **if** the induced subgraph of $S_j \cup S$ contains no copies of H **then**
12: **ACCEPT**
13: **end if**
14: **end for**
15: **end if**
16: **end for**
17: **REJECT**

Fig. 1. The ϵ-tester.

Definition 6. Let $0 < \epsilon < 1$, $N, k \in \mathbb{N}$, $\lambda = k/N$, H is a fixed digraph with h vertices, m be the number of edges in H, $\sigma = \binom{\binom{h}{2}}{m}$, $\kappa = \log_{1-\frac{(\sqrt{\epsilon})^h}{2}}(1/6) = \Theta(1/\epsilon^{h/2})$, and $\theta = \max\{\log_{\frac{6f(\epsilon;H)h!}{\sigma\lambda^2}}(2/3)^{1/\kappa}, 1\} = \Theta(f(\epsilon; H))$ when $f(\epsilon; H)$ is only dependent on $1/\epsilon$. If the value of $f(\epsilon; H)$ is large enough such that $\left(\frac{f(\epsilon;H)h!}{\binom{\binom{h}{2}}{m}\lambda}\right)^{\theta} \geq (\lambda/6)^{\theta}(2/3)^{1/\kappa}$, then we say $f(\epsilon; H)$ satisfies condition 1.

Fact 7 ([7]). *For a connected H, $f(\epsilon; H)$ has a polynomial dependency on $1/\epsilon$ if and only if the core of H is either an oriented tree or a directed cycle of length 2.*

By Fact 7, $f(\epsilon; H)$ has a polynomial dependency on $1/\epsilon$ in many H. Since the value of $f(\epsilon; H)$ is independent of h and m and $\left(\frac{2}{3}\right)^{1/(\theta\kappa)} \leq 1$, assuming $f(\epsilon; H) = O((1/\epsilon)^j)$, we can find a smaller $\epsilon = O\left(\left[\frac{\binom{\binom{h}{2}}{m}}{h!}\frac{\lambda^2}{6}\left(\frac{2}{3}\right)^{1/(\theta\kappa)}\right]^{-j}\right)$ such that

$$f(\epsilon; H) \geq \frac{\binom{\binom{h}{2}}{m}}{h!}\frac{\lambda^2}{6}\left(\frac{2}{3}\right)^{1/(\theta\kappa)}$$

i.e.,

$$\left(\frac{f(\epsilon; H)h!}{\binom{\binom{h}{2}}{m}\lambda}\right)^{\theta} \geq (\lambda/6)^{\theta}(2/3)^{1/\kappa};$$

hence $f(\epsilon; H)$ satisfies condition 1.

Claim 8. *Assume $0 < \epsilon < 1$, $N, k \in \mathbb{N}$ and $k \geq \sqrt{\epsilon}N$. Suppose the input digraph $G = (V, E)$ with N vertices contains an H-free k-induced subgraph, say $K = (V_k, E_k)$. The probability of $S \subseteq V_k$ for a random h-subset $S \subseteq V$ is greater than $(\sqrt{\epsilon})^h/2$ for N large enough.*

Proof. The probability of $S \subseteq V_k$ for a random h-set S is

$$\frac{\binom{k}{h}}{\binom{N}{h}} = \frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)}.$$

Since $k \geq \sqrt{\epsilon}N$, the above probability is at least

$$\frac{\sqrt{\epsilon}N(\sqrt{\epsilon}N-1)\cdots(\sqrt{\epsilon}N-h+1)}{N(N-1)\cdots(N-h+1)} > \frac{(\sqrt{\epsilon})^h}{2}$$

for N large enough. ■

Claim 9. *Let* $0 < \epsilon < 1$, $N, k \in \mathbb{N}$, $\lambda = k/N$, *H is a fixed digraph with h vertices and m be the number of edges in H. Suppose the input graph* $G = (V, E)$ *with N vertices is* ϵ*-far from any digraph having property* $P_{k,H}$*. The probability of finding an h-set whose induced subgraph contains copies of H is at least* $f(\epsilon; H)h!/\left[\binom{\binom{h}{2}}{m}\lambda\right]$.

Proof. By Corollary 4, each k-induced subgraph of G contains at least $f(\epsilon; H)N^h/\left[\binom{\binom{h}{2}}{m}\right]$ h-sets whose induced subgraphs contain copies of H. Therefore, by dividing V into N/k k-sets, we can find at least $\left[f(\epsilon; H)N^h/\binom{\binom{h}{2}}{m}\right](N/k) = f(\epsilon; H)N^h/\left[\binom{\binom{h}{2}}{m}\lambda\right]$ h-sets whose induced subgraphs contain copies of H in G, and the probability of finding an h-set whose induced subgraph contains copies of H is at least

$$\frac{\frac{f(\epsilon;H)N^h}{\binom{\binom{h}{2}}{m}\lambda}}{\binom{N}{h}} = \frac{f(\epsilon; H)N^h \cdot \frac{1}{\lambda} \cdot h!}{N(N-1)\cdots(N-h+1)\binom{\binom{h}{2}}{m}}$$
$$> \frac{f(\epsilon; H)h!}{\binom{\binom{h}{2}}{m}\lambda}. \quad \blacksquare$$

The following theorem proves the testability of $P_{k,H}$.

Theorem 10. *Let* $0 < \epsilon < 1$, $0 < k < N$ *is an integer and H is a fixed digraph. If* $f(\epsilon; H)$ *satisfies condition* 1*, the property* $P_{k,H}$ *is testable with a query complexity independent of the input size.*

Proof. Suppose $k < \sqrt{\epsilon}N$. Since $\binom{k}{2} < \epsilon\binom{N}{2}$, the number of edges in a k-induced subgraph is less than ϵN^2. The input graph G, therefore, cannot be ϵ-far from any digraph which has property $P_{k,H}$, and we can simply accept it. Assume $k \geq \sqrt{\epsilon}N$ for the rest of the proof.

Suppose the input digraph $G = (V, E)$ contains an H-free k-induced subgraph, say $K = (V_k, E_k)$. The probability that the algorithm accepts G is at least the probability of selecting a subset of V_k in step 6 of the algorithm and the tester accepts in step 12 for some j.

By Claim 8, the probability of $S \not\subseteq V_k$ is at most $1 - (\sqrt{\epsilon})^h/2$. As we independently select κ h-sets S, the probability of $S \not\subseteq V_k$ for all κ of them is at most $\left[1 - (\sqrt{\epsilon})^h/2\right]^\kappa = 1/6$. Assume $S \subseteq V_k$ from now on. We randomly select p other vertices (with replacements) in step 8. Denote the jth such (θh)-multiset by S_j. The algorithm then checks if the induced subgraph of $S_j \cup S$ contains a copy of H. Let event B mean $S_j \cup S$ contains a copy of H for all j. Given $S \subseteq V_k$, if more than θh vertices are selected from V_k in step 8, then event B will not occur (note that $\theta \geq 1$). Thus the probability of event B is at most the probability that the algorithm selects fewer than θh vertices from V_k in step 8. Let y be the number of vertices of these p vertices selected in step 8 that belong in V_k (with multiplicity counted). Then $\Pr[\text{ event } B\,] \leq \Pr[\,y < \theta h]$. We estimate the upper bound of the above probability by the Chernoff bound. As the probability of selecting a vertex in V_k is $k/N = \lambda$ and the total number of selections is $p = 6\theta h/\lambda$, we have $\mu = E[\,y\,] = (6\theta h/\lambda)\lambda = 6\theta h$. Rewrite $\Pr[\text{ event } B\,] = \Pr[\,y < (1-\Delta)6\theta h\,]$, where $\Delta = 5/6$. By the Chernoff bound, $\Pr[\text{ event } B\,] \leq e^{-\mu\Delta^2/2} = e^{-6\theta h(5/6)^2/2} = e^{-25\theta h/12}$. Since $\theta h > 1$, $\Pr[\text{ event } B\,] < e^{-2} < 1/6$. Hence the probability that we select an h-set from V_k in step 6 that leads to acceptance in step 12 is at least $(1 - 1/6)(1 - 1/6) > 2/3$. The probability that a digraph G which has property $P_{k,H}$ will be rejected is thus less than 1/3.

On the other hand, suppose the input graph $G = (V, E)$ is ϵ-far from any digraph which has property $P_{k,H}$. Obviously, the probability that the algorithm accepts is equal to the probability that we find an h-set S whose induced subgraph does not contain an H, and after we randomly select p additional vertices (with replacements), there exist a (θh)-multiset S_j from those p selected vertices such that the induced subgraph of $S_j \cup S$ contain no copies of H. By Claim 9, the probability of finding an h-set that contains copies of H is at least $f(\epsilon; H)h!/\left[\binom{\binom{h}{2}}{m}\lambda\right]$. For each (θh)-multiset S_j, at least θ disjoint h-sets are checked; hence the probability that $S_j \cup S$ contains copies of H is at least

$$\left(\frac{f(\epsilon; H)h!}{\binom{\binom{h}{2}}{m}\lambda}\right)^\theta = (\lambda/6)^\theta \cdot (2/3)^{1/\kappa}.$$

We then test $\binom{p}{\theta h}$ (θh)-multisets in step 12. Since

$$\binom{p}{\theta h} = \frac{(6\theta h/\lambda)!}{(\theta h)!} = \frac{(6\theta h/\lambda)[(6\theta h/\lambda) - 1]\cdots[(6\theta h/\lambda) - \theta h]}{(\theta h)!} > (6/\lambda)^{\theta h} > (6/\lambda)^\theta,$$

the probability that the induced subgraph of $S_j \cup S$ contains copies of H for all j is at least $(6/\lambda)^\theta \cdot (\lambda/6)^\theta \cdot (2/3)^{1/\kappa} = (2/3)^{1/\kappa}$. So, for each h-set S that passes the test in step 7, the probability that S does not lead to acceptance in step 12 is at least

```
if k < √ε N then
   ACCEPT
end if
let λ = k/N, γ = δh(h−1)/λ²
randomly select log_{(1−γ)λ^h} 2/3 h-subsets S from V
if there is an h-subset S whose induced subgraph does not contain a copy
of H then
   ACCEPT
else
   REJECT
end if
```

Fig. 2. The (ϵ, δ)-tester.

$(2/3)^{1/\kappa}$. Hence, regardless whether S passes the test in step 7, the probability that none of the S leads to acceptance in step 12 is at least $\left[(2/3)^{1/\kappa}\right]^{\kappa} = 2/3$. Therefore the probability that the algorithm accepts the input is less than 1/3.

The query complexity of step 7 is $O\left(h^2\right)$ and the query complexity from step 9 to step 10 is $O\left(\binom{p}{\theta h}\binom{\theta}{2}\right)$. Since $\binom{p}{\theta h}\binom{\theta h}{2} > h^2$, the query complexity is $O\left(\kappa\binom{p}{\theta h}\binom{\theta h}{2}\right)$. This value is independent of N. Hence the theorem follows. ■

4. An (ϵ, δ)-Tester

As in Section 3, let H be a fixed digraph with h vertices and m edges and G be the input digraph with N vertices. Observe that even if the input digraph $G = (V, E)$ is δ-close to having property $P_{k,H}$, G may contain no H-free k-induced subgraph. Thus we cannot simply select a subgraph S of G with size greater than h and then reject if S contains a copy of H. However, if G is δ-close to having property $P_{k,H}$, then it contains a k-induced subgraph $K = (V_k, E_k)$ such that K contains few, if any, copies of H. This is because if every k-induced subgraphs of G contain many copies of H, thus G cannot be δ-close to having property $P_{k,H}$. Let $L = \{M \mid M$ is an h-subset of V_k and M's induced subgraph contains at least a copy of $H\}$. From the above observation, we know that many h-subsets of V_k are not members of L. Thus if we select enough numbers of h-subsets from V, at least one of them is expected to be a subset of V_k and disjoint with M for all $M \in L$. On the other hand, suppose the input graph G is ϵ-far from any digraph which has property $P_{k,H}$. We expect that when we select reasonably many h-subsets from V, the induced subgraphs of all of them contain a copy of H. The (ϵ, δ)-tester that implements the above ideas is detailed in Fig. 2.

Claim 11. *Assume H is a fixed digraph with h vertices, $0 < \delta < 1$, $G = (V, E)$ is a digraph with N vertices, $0 < k < N$ is an integer, $\lambda = k/N$ and $\gamma = \frac{\delta h(h-1)}{\lambda^2}$. Let $K = (V_k, E_k)$ be a subgraph of G with k vertices such that at most $\delta\binom{N}{2}$ edges have to be removed from K to make it H-free and $U = \{W \mid W$ is an h-subset of V_k whose induced subgraph contains no copies of $H\}$. Then every time we select an h-subset S from V, the probability of $S \in U$ is at least $(1-\gamma)\lambda^h$.*

Proof. We can change at most $\delta\binom{N}{2}$ edges to make K H-free. These edges will be called bad edges. Since each bad edge connects two vertices and $|E_k| = k$, each bad edge is contained in at most $\binom{k-2}{h-2}$ copies of H in K. Thus, there are at most $\delta\binom{N}{2}\binom{k-2}{h-2}$ h-subsets of K whose induced subgraph contains a copy of H. Hence, every time we select an h-subset S from V, the probability of $S \in U$ is at least

$$
\begin{aligned}
\frac{\binom{k}{h} - \delta\binom{N}{2}\binom{k-2}{h-2}}{\binom{N}{h}} &= \frac{\frac{k!}{(k-h)!h!} - \delta\frac{N!}{(N-2)!2!}\frac{(k-2)!}{(k-h)!(h-2)!}}{\frac{N!}{(N-h)!h!}} \\
&= \frac{\frac{k(k-1)\cdots(k-h+1)}{h!} - \frac{\delta N(N-1)}{2}\frac{(k-2)(k-3)\cdots(k-h+1)}{(h-2)!}}{\frac{N(N-1)\cdots(N-h+1)}{h!}} \\
&= \frac{k(k-1)\cdots(k-h+1)}{N(N-1)\cdots(N-h+1)} - \frac{\delta(k-2)(k-3)\cdots(k-h+1)h(h-1)}{2(N-2)\cdots(N-h+1)}.
\end{aligned}
$$

As c is a constant, $\lim_{N\to\infty}\frac{k-c}{N-c} = \lambda$. As

$$
\begin{aligned}
\lim_{N\to\infty}\frac{\binom{k}{h} - \delta\binom{N}{2}\binom{k-2}{h-2}}{\binom{N}{h}} &= \lambda^h - \frac{\delta}{2}\lambda^{h-2}h(h-1) = \left(1 - \frac{\delta h(h-1)}{2\lambda^2}\right)\lambda^h \\
&= (1-\gamma/2)\lambda^h > (1-\gamma)\lambda^h,
\end{aligned}
$$

the probability of $S \in U$ is at least $(1-\gamma)\lambda^h$ with N large enough. ■

Definition 12. Let $N, k \in \mathbb{N}$, $\lambda = k/N$, $\gamma = \frac{\delta h(h-1)}{\lambda^2}$, H be a fixed digraph with h vertices and m be the number of edges in H. If λ, h and δ satisfy $\log_{(1-\gamma)\lambda^h} 2/3 > 0$ and if $f(\epsilon; H)$ is such that $\left[1 - \frac{f(\epsilon;H)h!}{\binom{\binom{h}{2}}{m}\lambda}\right] < \frac{(1-\gamma)\lambda^h}{2^{1/\log_{(1-\gamma)\lambda^h} 2/3}}$, then we say $P_{k,H}$ and $f(\epsilon; H)$ satisfy condition 2.

By Fact 7, $f(\epsilon; H)$ has a polynomial dependency on $1/\epsilon$ for many H. Suppose $P_{k,H}$ satisfies condition 2. Since λ, h and δ are independent of ϵ, no matter what the value of $\left[1 - \frac{(1-\gamma)\lambda^h}{2^{1/\log_{(1-\gamma)\lambda^h} 2/3}}\right] \frac{\binom{\binom{h}{2}}{m}\lambda}{h!}$ is, we can find a smaller ϵ such that the value of $f(\epsilon; H)$ is greater than $\left[1 - \frac{(1-\gamma)\lambda^h}{2^{1/\log_{(1-\gamma)\lambda^h} 2/3}}\right] \frac{\binom{\binom{h}{2}}{m}\lambda}{h!}$ and thus satisfies condition 2.

Theorem 13. *Let $0 < \epsilon, \delta < 1$, k, N be integers, $0 < k < N$ and H be a fixed digraph. For $P_{k,H}$ and $f(\epsilon; H)$ satisfying condition 2 and assume the input is a digraph with N vertices, the property $P_{k,H}$ is (ϵ, δ)-testable with a query complexity independent of the input size.*

Proof. As in the proof of Theorem 10, we accept the input if $k < \sqrt{\epsilon}N$. Now assume $k \geq \sqrt{\epsilon}N$ for the rest of the proof. Suppose the input graph $G = (V, E)$ is δ-close to a digraph which has property $P_{k,H}$. Then G must contain a k-induced subgraph $K = (V_k, E_k)$ such that we can change at most $\delta\binom{N}{2}$ edges to make K H-free. Just like the proof of Theorem 10, the probability that the algorithm accepts G is at least the probability of selecting $\log_{(1-\gamma)\lambda^h} 2/3$ h-subsets of V in step 5 of the algorithm and at least one of them whose induced subgraph contains no copies of H.

After we independently select $\log_{(1-\gamma)\lambda^h} 2/3$ h-subsets from V, by Claim 11, the probability of selecting at least one $S \in U$ is $[(1-\gamma)\lambda^h]^{\log_{(1-\gamma)\lambda^h} 2/3} = 2/3$. It means that G will be accepted with probability at least $2/3$.

On the other hand, suppose the input graph $G = (V, E)$ is ϵ-far from any digraph having property $P_{k,H}$. The probability that the algorithm accepts G is the probability that we find at least an h-subset whose induced subgraph does not contain an H in step 5 of the algorithm. By Claim 9, the probability of finding an h-subset that contains no copies of H is at most $1 - \frac{f(\epsilon;H)h!}{\binom{\binom{h}{2}}{m}\lambda}$. Since $f(\epsilon; H)$ satisfies condition 2, the probability of selecting at least one H-free h-subset is at most

$$\begin{aligned}
\left[1 - \frac{f(\epsilon; H)h!}{\binom{\binom{h}{2}}{m}\lambda}\right]^{\log_{(1-\gamma)\lambda^h} 2/3} &< \left[\frac{(1-\gamma)\lambda^h}{2^{1/\log_{(1-\gamma)\lambda^h} 2/3}}\right]^{\log_{(1-\gamma)\lambda^h} 2/3} \\
&= \left[(1-\gamma)\lambda^h\right]^{\log_{(1-\gamma)\lambda^h} 2/3} \left[\frac{1}{2^{1/\log_{(1-\gamma)\lambda^h} 2/3}}\right]^{\log_{(1-\gamma)\lambda^h} 2/3} \\
&= \frac{2}{3} \cdot \frac{1}{2} = 1/3.
\end{aligned}$$

It means that G will be rejected with probability at least $2/3$.

It is clear that the query complexity is independent of N, as in the proof of Theorem 10. ∎

Property $P_{k,H}$ may not satisfy condition 2 for some k and H. As our (ϵ, δ)-tester only works when condition 2 is satisfied, compared with the ϵ-tester in Section 3, there are different restrictions on using the (ϵ, δ)-tester. Therefore, we cannot simply use an $(\epsilon, 0)$-tester as an ϵ-tester.

5. Conclusion

We prove that there is no efficient algorithm that can determine whether a digraph contains an H-free k-induced subgraph. Then we create two property testing algorithms to supply an approximate answer with constant query complexities. The results hold with the following oracle: we can specify any adjacency matrix of a digraph G and ask whether an edge exists between any pair of vertices. The query complexities of our property testing algorithms are independent of the input digraph's size.

References

[1] Ronitt Rubinfeld, Madhu Sudan, Robust characterizations of polynomials with applications to program testing, SIAM J. Comput. 25 (2) (1996) 252–271.
[2] Richard C. Holt, Edward M. Reingold, On the time required to detect cycles and connectivity in graphs, Math. Syst. Theory 6 (2) (1972) 103–106.
[3] Béla Bollobás, Complete subgraphs are elusive, J. Combin. Theory Ser. B 21 (1976) 1–7.
[4] Catherine E. Chronaki, A survey of evasiveness: Lower bounds on the decision-tree complexity of boolean functions, Available on: http://www.ics.forth.gr/~chronaki/papers/ur/eve.ps, 2002.
[5] David G. Kirkpatrick, Determining graph properties from matrix representations, in: STOC, 1974, pp. 84–90.
[6] Ronald L. Rivest, Jean Vuillemin, A generalization and proof of the Aanderaa–Rosenberg conjecture, in: STOC, 1975, pp. 6–11.
[7] Noga Alon, Asaf Shapira, Testing subgraphs in directed graphs, in: STOC, 2003, pp. 700–709.

[8] Oded Goldreich, Shafi Goldwasser, Dana Ron, Property testing and its connection to learning and approximation, J. ACM 45 (4) (1998) 653–750.
[9] Noga Alon, Asaf Shapira, A characterization of easily testable induced subgraphs, in: SODA, 2004, pp. 942–951.
[10] Noga Alon, Eldar Fischer, Michael Krivelevich, Mario Szegedy, Efficient testing of large graphs, in: FOCS, 1999, pp. 656–666.
[11] Noga Alon, Eldar Fischer, Ilan Newman, Asaf Shapira, A combinatorial characterization of the testable graph properties: It's all about regularity, in: STOC, 2006, pp. 251–260.
[12] Oded Goldreich, Dana Ron, A sublinear bipartiteness tester for bounded degree graphs, Combinatorica 19 (3) (1999) 335–373.
[13] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, Alex Samorodnitsky, Improved testing algorithms for monotonicity, in: RANDOM-APPROX, 1999, pp. 97–108.
[14] Funda Ergun, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, Mahesh Viswanathan, Spot-checkers, in: STOC, 1998, pp. 259–268.
[15] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, Alex Samorodnitsky, Testing monotonicity, Combinatorica 20 (3) (2000) 301–337.
[16] Michael J. Kearns, Dana Ron, Testing problems with sublearning sample complexity, J. Comput. System Sci. 61 (3) (2000) 428–456.
[17] Michal Parnas, Dana Ron, Ronitt Rubinfeld, Tolerant property testing and distance approximation, Electron. Colloq. Comput. Complex. (ECCC) 11 (10) (2004).
[18] Eldar Fischer, Lance Fortnow, Tolerant versus intolerant testing for boolean properties, Electron. Colloq. Comput. Complex. (ECCC) 11 (105) (2004).
[19] Eldar Fischer, Ilan Newman, Testing versus estimation of graph properties, in: STOC, 2005, pp. 138–146.
[20] Tamas Kövari, Vera T. Sós, Paul Turán, On a problem of K. Zarankiewicz, Colloq. Math. 3 (1954) 50–57.
[21] Uriel Feige, Shimon Kogan, The hardness of approximating hereditary properties. Available on: http://research.microsoft.com/research/theory/feige/homepagefiles/hereditary.pdf, 2005.