Every property is testable on a natural class of scale-free multigraphs

Hiro Ito*

April 6, 2015

Abstract

In this paper, we introduce a natural class of multigraphs, Hierarchical-Scale-Free (HSF, in short), and consider constant-time testability on the class. We show that a very wide subclass, i.e., the power-low exponent is greater than two, of HSF is hyperfinite. Based on this result, an algorithm of deterministic partitioning oracle can be constructed. And finally we show that every property is constant-time testable on the above subclass of HSF. This algorithm utilizes the result of Newman and Sohler of STOC’11. However, their algorithm is on the bounded-degree model, while it is known that the real scale-free networks usually includes some hubs, which have very large degree. HSF is based on the scale-free property and includes such hubs. This is the first universal result of constant-time testability on the general graph model, and it has a potential to be applicable on very wide real scale-free networks.

1 Introduction

Nowadays how to treat big data is one of the most important problem in the area of computer science. For theoretical area, developing efficient algorithms for treating big data is an urgent task. In this purpose, sublinear-time algorithms look powerful tools, since they are enough to read very small part (constant size) of the inputs.

Property testing is the most well studied in the area of sublinear-time algorithms. A testing algorithm (or a tester) for a property accepts an input if it has the property and rejects it if it’s far from having the property in high probability (e.g., at least 2/3) by reading a constant part of the input. A property is said to be testable if there is a tester[9].

Property testing on graph properties has been well studied and many fruitful results have been obtained[2, 3, 9, 10, 11, 12, 17, 19]. Testers on graphs are separated into three groups from the view point of models: the dense graph model (the adjacent-matrix model), the bounded-degree model, and the general model. The dense graph model is the best clarified: In this model, a characterization of testable properties was obtained[2]. However, graphs based on real models are usually sparse and unfortunately the dense graph model does not work. Studies on the bounded-degree model proceeds recently. One of the most important result on this model is that every minor-closed property is testable[3]. This result is extended to a surprising result that every property on hyperfinite graphs is testable[17]. However, graphs based on real models have no degree bounds, i.e., it is known that web-graphs have some hubs[1, 15], which have very large degree, and unfortunately again, these algorithms don’t work for them.

*School of Informatics and Engineering, The University of Electro-Communications (UEC), Tokyo, Japan; and CREST, JST, Tokyo, Japan; itohiro@uec.ac.jp
A typical big-data graph model is “scale-free networks.” Many models have been proposed for scale-free networks \cite{1, 4, 8, 15, 16, 18, 20, 21, 22}. Recently a promising model based on another property of hierarchical isomorphic structure is presented: If we see a graph in broad perspective, we find a similar structure to local structures. Shigezumi, Uno, and Watanabe\cite{20} presented a model, which is based on an idea of “hierarchical isomorphic structure of power-low distribution of isolated cliques,” and it looks one of the most persuasive models for representing web-graphs. Isolated cliques was presented by Ito and Iwama \cite{13, 14} and the definition is as follows. For a nonnegative integer \(c \geq 0\), an \(c\)-isolated clique is a clique such that the number of outgoing edges (edges between the clique and the other vertices) is less than \(ck\), where \(k\) is the number of vertices of the clique. A 1-isolated clique is sometimes said as an isolated clique, simply.

Based on the model of \cite{20}, we introduce a class of multigraphs, Hierarchical Scale-Free Multigraphs (HSF, in short, Definitions \cite{2, 5}), which represents natural scale-free networks. And we show the following result (Theorem \cite{2, 5}):

Every property is testable on HSF if the power-low exponents are greater than two.

From this result, many problems on real scale-free big networks will be solvable in constant-time. Although this result is a kind of an application of the algorithms of \cite{17}, which is a result on bounded-degree graphs, HSF is not a class of bounded-degree graphs. This is the first result on universal algorithms for the general graph model.

2 Preliminary

2.1 Definitions

In this paper, we consider undirected multigraphs without self-loops. We call such a multigraph a graph simply in this paper and use \(G = (V, E)\) to denote it, where \(V\) is the vertex set and \(E\) is the edge set.

For a graph \(G = (V, E)\) and vertex subsets \(X, Y \subseteq V\), \(E_G(X,Y)\) denotes the edge set between \(X\) and \(Y\), i.e., \(E_G(X,Y) = \{(x, y) \in E \mid x \in X, y \in Y\}\). \(E_G(X, V - X)\) is also written as \(E_G(X)\) simply. \(|E_G(X)|\) are denoted by \(d_G(X)\). For a vertex \(v \in V\), the number of edges incident to \(v\) it called the degree of \(v\). A singleton set \(\{v\}\) may often be written as \(x\) for notational simplicity. E.g., the degree of \(v\) is represented by \(d_G(v)\). The subscript \(G\) of the above \(E_G(\cdot, \cdot), d_G(\cdot, \cdot)\) etc. may be omitted if it is clear. For a graph \(G = (V, E)\) and a vertex subset \(X \subseteq V\), the subgraph induced by \(X\) is defined as \(G(X) = (X, \{(u, v) \in E \mid u, v \in X\})\).

For a vertex subset \(X \subseteq V\), an operation to replace \(X\) with a new vertex \(v_X\) and each edge incident to a vertex in \(X\) with a new edge incident to \(v_X\) is called a contraction of \(X\). That is, by contracting \(X \subseteq V\), a graph \(G = (V, E)\) is changed to \(G' = (V', E')\) such that

\[
V' = V - X \cup \{v_X\}, \quad E' = E - \{(u,v) \mid v \in X, u \in V - X\} \cup \{(v_X, u) \mid u \in V - X\}.
\]

We identify the above \((v_X, u)\) with \((v, u)\). In other word, we say that \((v, u)\) remains in \(G'\) (as \((v_X, u)\)). Note that graphs are multigraphs and thus if there are two edges \((v, u), (v', u) \in E\) for \(v, v' \in X\) and \(v \neq v'\), then two parallel edges both represented by \((v_X, u)\), one of which corresponds to \((v, u)\) and the other corresponds to \((v', u)\), are added to \(E'\).

Two graphs \(G_1 = (V_1, E_1)\) and \(G_2 = (V_2, E_2)\) are isomorphic if there is a one to one correspondence \(\Phi : V_1 \to V_2\) such that \(|E_{G_1(u,v)}| = |E_{G_2}(\Phi(u), \Phi(v))|\) for all \(u, v \in V_1\). A graph property (or property, in short) is a (possibly infinite) family of graphs, which is closed under isomorphism.
Definitions 2.1 (ε-far) Let $G = (V, E)$ and $G' = (V', E')$ be two graphs with $|V| = |V'| = n$ vertices. Let $m(G, G')$ be the amount of edges that needs to be deleted and/or inserted from $G$ in order to make it isomorphic to $G'$. The distance between $G$ and $G'$ is defined as $\text{dist}(G, G') = m(G, G')/n$. We say that $G$ and $G'$ are $\varepsilon$-far if $\text{dist}(G, G) > \varepsilon$, and $\varepsilon$-close, otherwise. Let $P$ be a non-empty property. The distance between $G$ and $P$ is $\text{dist}(G, P) = \min_{G'' \in P} \text{dist}(G, G'')$. We say that $G$ is $\varepsilon$-far from $P$ if $\text{dist}(G, P) > \varepsilon$, and $\varepsilon$-close, otherwise.

A property testing algorithm for a property $P$, or tester, for short, is an algorithm that given query access to a graph $G$, accepts every graph from $P$ with probability at least $2/3$, and reject every graph that is $\varepsilon$-far from $P$ with probability at least $2/3$. A query in the general graph model is for any vertex $v$, the algorithm may ask for the degree $d(v)$, and it may ask to get the $i$th neighbor of the vertex for every $1 \leq i \leq d(v)$.

Definitions 2.2 (isolated cliques) For a graph $G = (V, E)$ and a real number $c \geq 0$, a vertex subset $Q \subseteq V$ is called a $c$-isolated clique if $Q$ is a clique (i.e., $(u, v) \in E$, $\forall u, v \in Q$ and $u \neq v$) and $d_G(Q) < c|Q|$. A 1-isolated clique is sometimes called an isolated clique simply. $E(G)$ is the graph obtained from $G$ by contracting all isolated cliques.

Note that $E(G)$ is uniquely defined, since distinct two isolated cliques are never overlapped from the definition of isolated cliques.

Definitions 2.3 (hyperfinite) For real numbers $t > 0$ and $\varepsilon > 0$, a graph $G = (V, E)$ consisting of $n$ vertices is $(t, \varepsilon)$-hyperfinite if one can remove at most $en$ edges from $G$ and obtain a graph whose connected components have size at most $t$. For a function $\rho: R^+ \rightarrow R^+$, $G$ is $\rho$-hyperfinite if it is $(\rho(\varepsilon), \varepsilon)$-hyperfinite for all $\varepsilon > 0$. A family $\mathcal{G}$ of graphs is $\rho$-hyperfinite if all $G \in \mathcal{G}$ are $\rho$-hyperfinite. A family $\mathcal{G}$ of graphs is hyperfinite if there exists a function $\rho$ such that $\mathcal{G}$ is $\rho$-hyperfinite.

Hyperfinite is a large class, since it is known that any minor-closed property is hyperfinite in the bounded degree model. From the view point of testing, the importance of hyperfiniteness comes from the following result.

Theorem 2.4 ([17]) For the bounded degree model, any property is testable for any class of hyperfinite graphs.

This result is very strong. But there is a problem that the result world on bounded-degree graphs and natural graph class that models real scale-free networks don’t have a degree bound. We formalize two natural classes $\mathcal{SF}$ and $\mathcal{HSF}$ that represent scale-free networks. The latter is a subclass of the former.

Definitions 2.5 For positive real numbers $c_1, c_2 > 1$ and $\gamma_1 > 1$, a class of Scale-Free Graphs (SF, in short) $\mathcal{SF} = \mathcal{SF}(c_1, c_2, \gamma_1)$ consists of (multi)graphs $G = (V, E)$ the following condition hold:

(i) Let $\nu_i$ be the number of vertices $v$ with $d(v) = i$. Then

\[ c_1i^{-\gamma_1}n \leq \nu_i \leq c_2i^{-\gamma_1}n, \quad \forall i \in \{2, 3, \ldots\}. \]  

1 The distance may be larger than 1, since $m(G, G') > n$ may occurs. However, we consider sparse graphs and they have a upper bound of the average (not possibly maximum) degree, say $d$, and thus $\text{dist}(G, G')$ is bounded by $d/2$.

2 Although to ask whether there is an edge between any two vertices is also allowed in the general graph model, the algorithms we use in this paper do not need to use this query.
The above property (i) is generally called “power-low” and in many real scale-free networks, it is said that $2 < \gamma_1 < 3$.\[1\]

**Definitions 2.6 (Hierarchical Scale-Free Graphs)** For positive real numbers $c_1, c_2, c_3, c_4 > 1$ and $\gamma_1, \gamma_2 > 1$, a class of Hierarchical Scale-Free Graphs (HSF in short) $\mathcal{H}_\infty = \mathcal{H}_\infty(c_1, c_2, c_3, c_4, \gamma_1, \gamma_2)$ consists of (milti)graphs $G = (V, E)$ such that all the following conditions hold:

(i) $G \in \mathcal{S}_\infty(c_1, c_2, \gamma_1)$

(ii) Let $\mu_i$ be the number of isolated cliques $Q$ with $|Q| = i$. Then

$$[c_3 i^{-\gamma_2} n] \leq \mu_i \leq c_4 i^{-\gamma_2} n, \quad \forall i \in \{2, 3, \ldots, \},$$  \hspace{1cm} (2)

(iii) $E(G) \in \mathcal{H}_\infty$.

The above (ii) and (iii) are pointed by [20]. It indicates that $3 < \gamma_2 < 4$ in many real scale-free networks.

### 2.2 Our Contribution

We show the following results.

**Theorem 2.7** For any $\mathcal{H}_\infty = \mathcal{H}_\infty(c_1, c_2, c_3, c_4, \gamma_1, \gamma_2)$ with $\gamma_1, \gamma_2 > 2$ and any real number $\epsilon > 0$, there is a real number $t_\epsilon = t_\epsilon(\mathcal{H}_\infty, \epsilon)$ such that $\mathcal{H}_\infty$ is $(t_\epsilon, \epsilon)$-hyperfinites.

We give a global algorithm for obtaining the partition realizing the hyperfinites of Theorem 2.7. The algorithm is deterministic, i.e., if a graph and the parameter $\epsilon$ are fixed, then the partition is also fixed. The algorithm can be easily revised to an local algorithm and we get a deterministic partitioning oracle to get the partition (Lemma 4.2). Note that all known partitioning algorithms are randomized algorithms. By using this partitioning oracle and a discussion similar to one used in [17], we get the following main theorem.

**Theorem 2.8** Any property is testable for $\mathcal{H}_\infty(c_1, c_2, c_3, c_4, \gamma_1, \gamma_2)$ with $\gamma_1, \gamma_2 > 2$.

### 3 Hyperfiniteneses and a Global Partitioning Algorithm

For a graph $G$ and a nonnegative integer $d \geq 0$, $G|d$ is a graph made by deleting all edges insist to every vertex $v$ with $d(v) > d$ from $G$. Note that $G|d$ is a bounded-degree graph with degree bound $d$.

**Lemma 3.1** For any $\mathcal{S}_\infty = \mathcal{S}_\infty(c_1, c_2, \gamma_1)$ with $\gamma_1 > 2$, and any positive real number $\epsilon > 0$, there is a constant $t_\epsilon = t_\epsilon(\mathcal{S}_\infty, \epsilon)$ such that for any graph $G \in \mathcal{S}_\infty$, $G|t_\epsilon$ is $\epsilon$-close to $G$.

Before showing a proof of this lemma, we introduce some definitions. Riemann zeta function is defined by

$$\zeta(\gamma) = \sum_{i=1}^{\infty} i^{-\gamma}.$$  

This function is known to be converged to a constant $(\zeta(\gamma) < 1 + (1 - \gamma)^{-1})$ for any $\gamma > 1$. We introduce a generalization of this function by using a positive integer $k \geq 1$ as follows.

$$\zeta(k, \gamma) = \sum_{i=k}^{\infty} i^{-\gamma}.$$  

Note that $\zeta(\gamma) = \zeta(1, \gamma)$.  

4
Lemma 3.2 For any $\epsilon > 0$ and $\gamma > 1$, there is an integer $k_{3.2} = k_{3.2}(\epsilon, \gamma) \geq 1$ such that 
$$\zeta(k_{3.2} \gamma) < \epsilon.$$  

Proof: Clear from the fact that $\zeta(\gamma)$ is converged for every $\gamma > 1$. $\square$

Proof of Lemma 3.2 Let $d$ be an arbitrary positive integer. Let $m_d$ be the number of removed edges to make $G\mid d$ from $G$. From (1),

$$m_d = \sum_{i=d+1}^{\infty} i\nu_in \leq \sum_{i=d+1}^{\infty} c_2i^{-((\gamma_1-1))}n = c_2\zeta(d+1,\gamma_1-1)n.$$  

From the assumption of $\gamma_1 > 2$ and Lemma 3.2, $\zeta(d+1,\gamma_1-1) < \epsilon/c_2$ if $d+1 \geq k_{3.2}^c(\epsilon/c_2, \gamma_1-1)$. Thus by letting $\delta = \delta_{3.1}(\epsilon, c_2, \gamma_1) = k_{3.2}^c(\epsilon/c_2, \gamma_1-1) - 1$, we have $m_d < \epsilon n$. $\square$

From here, we denote the above $\delta_{3.1}(\epsilon, c_2, \gamma_1)$ by $\delta$ for notational simplicity.

Lemma 3.3 For a graph $G \in \mathcal{HSF}(c_1, c_2, c_3, c_4, \gamma_1, \gamma_2)$, if $|V| \geq 2^{\gamma_2}/c_3$, then there is at least one isolated clique with size at least two in $G$.

Proof: From (2), $\mu_2 \geq |c_32^{-\gamma_2}n| \geq |c_32^{-\gamma_2}2^{\gamma_2}/c_3| = 1$. $\square$

Let $W_1, \ldots, W_k$ ($W_i \subseteq V$, $\forall i \in \{1, \ldots, k\}$) be a family of subsets of vertices satisfying that $W_i \cap W_j = \emptyset$ for every $i, j \in \{1, \ldots, k\}$ and $i \neq j$, and $W_1 \cup \cdots \cup W_k = V$. Then $P = \{W_1, \ldots, W_k\}$ is called a partition of $V$. In the followings, we explain a global algorithm for getting a partition of $V$ realizing the hyperfiniteness of a graph in $\mathcal{HSF}$. First we give a base algorithm of it.

procedure HierarchicalContraction($G$)  
begin  
  1 \hspace{1em} i := 0, G_0 := G  
  2 \hspace{2em} while there exists an isolated clique in $G$ do  
  3 \hspace{3em} i := i + 1, G_i := E(G_{i-1})  
  4 \hspace{2em} enddo  
end.

We denote $G_i = (V_i, E_i)$ for $i \in \{0, 1, \ldots\}$ and $N = 2^{\gamma_2}/c_3$. Let $G_k = (V_k, E_k)$ be the final graph of HierarchicalContraction($G$). From Lemma 3.3 $|V_k| < N$.

The trail of the contraction can be represented by a rooted tree $T = (V_T, E_T)$ defined as follows. Each leaf of $T$ corresponds to each vertex in $V_0$. If a clique $Q$ is contracted into $vQ$, add a vertex $v$ to $T$ as the parent of all vertices $v \in Q$. Every contraction operation is represented by such an addition of a parent and we get $|V_k|$ separated trees $T_v$ whose root is $v \in V_k$. Finally we add a root $r$ as the parent of each root of the above $T_v$.

For each vertex $x \in V_T$, we denote the set of vertices of $G_0$ contracted in $x$ by $W(x)$. We also define the weight of $x$ as $w(x) = |W(x)|$. $W(*)$ and $w(*)$ are also applied for vertices of $G_i$ ($i \in \{0, \ldots, k\}$) in the same way, i.e., for $v \in V_i$, $W(v)$ denotes the set of vertices in $V_0$ contracted in $v$. For any vertex subset $S \subseteq V_i$, $W(S) := \cup_{v \in S} W(v)$ and $w(S) := \sum_{v \in S} w(v)$.

For any $\epsilon > 0$, a vertex $x \in V_T$ is called a large vertex with respect to $\epsilon$ (or large vertex, in short) if $w(x) > 2\delta/\epsilon$. (Remember that $\delta = \delta_{3.1}(\epsilon, c_2, \gamma_1)$ defined in Lemma 3.1) From the definition, if a vertex is large, then the parent of the vertex is also large (and thus all ancestors of a large vertex are also large). If $x \in V_T$ is a large vertex, $W(x)$ is called a large component (with respect to $\epsilon$).

We color edges in $E$ ($= E_0$) in the following manner:
• For every vertex \( v \in V \) with \( d_G(v) > \delta \) \( (= \delta_1(\epsilon, c_2, \gamma_1)) \), every edge incident to \( v \) is colored by red.

• For every large component \( S \), for every edge \( e \in E_G(S) \), if \( e \) is not colored by red, then it is colored by blue.

• For every edge \( e \) that remains in \( E_k \), if it is not colored by neither red nor blue, then it is colored by yellow.

The other edges in \( E \) are not colored. The set of red, blue, and yellow edges in \( E \) are represented by \( R \), \( B \), and \( Y \), respectively. These colors are preserved in \( G_1 = E(G_0) \), \( G_2 = E(G_1) \), \ldots, \( G_k = E(G_{k-1}) \), e.g., if an edge \( e \in E_i \) is red, then the corresponding edge in \( E_{i+1} \) is also red.

**Lemma 3.4** For any \( G_i (i \in \{0, \ldots, k\}) \), all edges incident to a vertex with degree higher than \( \delta \) are red.

**Proof:** For \( G_0 = G \), the statement clearly holds from the rule of the coloring. Assume that the statement holds in \( G_{i-1} \) and does not hold in some \( G_i \). Let \( v \) be a vertex in \( V_i \) such that \( d_{G_i}(v) \geq \delta + 1 \) and a non-red edge is incident to \( v \). Then \( v \) must be made by contracting an isolated clique in \( G_{i-1} \), say \( Q \subseteq V_{i-1} \), such that \( d_{G_{i-1}}(Q) \geq \delta + 1 \). From the definition of isolated cliques, \( |Q| \geq d_{G_{i-1}}(Q) + 1 \geq \delta + 2 \). Since \( Q \) is a clique, every vertex \( Q \) has degree at least \( |Q| - 1 \geq \delta + 1 \) in \( G_{i-1} \). It follows that all edges incident to a vertex in \( Q \) must be red. This contradicts the assumption that a non-red edge is incident to \( v \). \( \square \)

**Lemma 3.5** The following properties hold:

1. \( |R|, |B| < \epsilon n \),
2. \( |Y| < \delta 2^{\gamma_2^{-1}} / c_3 \).

**Proof:**

1. \( |R| < \epsilon n \) is directly obtained from Lemma 3.1. Let \( S \subseteq V \) be a large component such that a non-red edge exists in \( E(S) \). From Lemma 3.4, \( d(S) \leq \delta \). Thus \( d(S)/w(S) < \delta / (2\delta/\epsilon) = 2\epsilon \). This means that the average degree of blue edges is less than \( 2\epsilon \). Therefore \( |B| < 2\epsilon n / 2 = \epsilon n. \)

2. From Lemma 3.4, all edges incident to a vertex with degree higher than \( \delta \) are red. From this, it follows that the number of non-red edges in \( E_k \) is at most \( \delta |V_k|/2 \). Thus the number of yellow edges in \( E \) is also at most \( \delta |V_k|/2 \). By considering \( |V_k| < N = 2^{\gamma_2}/c_3 \), we have \( |Y| < \delta 2^{\gamma_2^{-1}} / c_3 \).

Now we can prove Theorem 2.7.

**Proof of Theorem 2.7** If \( n \leq \delta 2^{\gamma_2^{-1}} / c_3 \epsilon \), then the statement is clear by setting as \( t \geq \delta 2^{\gamma_2^{-1}} / c_3 \epsilon \). Thus we assume that \( n > \delta 2^{\gamma_2^{-1}} / c_3 \epsilon \). Let \( G' \) be a graph obtained by deleting red, blue, and yellow edges from \( G \). From Lemma 3.5, the number of deleted edges is less than

\[ 2\epsilon n + \delta 2^{\gamma_2^{-1}} / c_3 < 3\epsilon n. \]  

Next, we will show that the maximum size of connected components in \( G' \) is at most \( 2\delta / \epsilon \). Assume that there exists a connected component \( G'(X) = (X, E_X) \) of \( G' \) consisting of more than \( 2\delta / \epsilon \) vertices. \( X \) does not include neither
• vertices $v$ with $d_G(v) > \delta$ (since all edges in $E_G(v)$ are colored by red) nor

• any other large components $Y$ ($\neq X$) (since all edges in $E_G(Y)$ are colored by blue or red).

Moreover, every edge included in $E_X$ is not colored by yellow. This means that these edges are contracted into a clique in a process of HierarchicalContraction$(G)$. Let $Q \subseteq V_i$ be the maximal one of such isolated cliques. From the maximality, $X \subseteq W(Q)$. Let $C \subseteq Q$ be the set of vertices of $V_i$ such that $C = \{v \in Q \mid w(v) \subseteq X\}$.

Here $X \subseteq W(C)$, since otherwise, $\exists v \in X - W(C)$ and from the connectivity of $G(X)$, there is a vertex $u \in W(C)$ such that edge $(u, v)$ is non-colored, contradicting to the maximality of $Q$. Moreover, we can observe that $|C| \geq 2$ and that vertices in $C$ are connected by non-colored edges in $G_i$, since otherwise, no non-colored edge in $E_X$ is removed by contracting $Q$ and it contradicting to the definition of $Q$.

For $v \in C$, $d_{G_i}(v) \leq \delta$ and $w(v) < 2\delta/\epsilon$, since otherwise, all the edges in $E(v)$ is colored by blue (or red) and contradicting to the former observation. From these discussions, $|C| \leq |Q| \leq \delta + 1$. Therefore

$$w(X) \leq w(C) = \sum_{v \in C} w(v) < (\delta + 1) \cdot \frac{2\delta}{\epsilon} = \frac{2\delta(\delta + 1)}{\epsilon},$$

contradiction.

Thus we proved that $G$ is $(\max\{\delta 2^{\gamma_2 - 1}/c_3 \epsilon, 2\delta(\delta + 1)/\epsilon\}, 3\epsilon)$-hyperfinite. Here, $\epsilon$ is arbitrary real number in $(0, 1]$, then by defining as $\frac{2\gamma_2}{\epsilon} = \max\{3\delta 2^{\gamma_2 - 1}/c_3 \epsilon, 6\delta(\delta + 1)/\epsilon\}$, $G$ is $(\frac{2\gamma_2}{\epsilon})$-hyperfinite for any $\epsilon > 0$. $\square$

4 Testing Algorithm

4.1 Deterministic partitioning oracle

The global partitioning algorithm of Theorem [2.7] can be easily revised to run locally, i.e., a “partitioning oracle” base on this algorithm can be obtained. A partitioning oracles, which calculates a partition realizing the hyperfiniteness locally, are introduced by Benjamini, et al. [3] implicitly and by Hassidim, et al. [12] explicitly. It is a powerful tool for constructing constant-time algorithms for sparse graphs. It is revised by some researchers and Levi and Ron’s algorithm [19] is the fastest so far. All algorithms for partitioning oracles presented so far are randomized algorithms. Our algorithm, however, does not use any randomized valuables and it runs deterministically for the partition. That is, we call it a deterministic partitioning oracle, which is rigorously defined as follows:

Definitions 4.1 $O$ is a deterministic $(t, \epsilon)$-partitioning oracle for a class of graphs $C$, if, given query access to a graph $G = (V, E)$, it provides query access to a partition $P$ of $G$. For a query about $v \in V$, $O$ returns $P(v)$. The partition has the following property:

• $P$ is a function of $G$, $t$, and $\epsilon$. (It does not depend on the order of queries to $O$.)

• For every $v \in V$, $|P(v)| \leq t$ and $P(v)$ induces a connected subgraph of $G$.

• If $G \in C$, then $|(u, v) \in E \mid P(u) \neq P(v)| \leq \epsilon|V|$.

Lemma 4.2 There is a deterministic $(\frac{2\gamma_2}{\epsilon})$-partitioning oracle $O_{HSF}$ for HSF with $\gamma_1, \gamma_2 > 2$ with query complexity $\delta^{O(\delta^2/\epsilon)}$ for one query.
Before giving a proof of this lemma, we introduce some notations as follows. A connected graph $G = (V, E)$ with a specified marked vertex $v$, is called a rooted graph and we sometimes say that $G$ is rooted at $v$. A rooted graph $G = (V, E)$ has a radius $t$, if every vertex in $V$ has distance at most $t$ from the root $v$. Two rooted graphs are isomorphic if there is a graph isomorphism between these graphs that identifies the roots with each other. We denote by $N(d, t)$ the number of all non-isomorphic rooted graphs with maximum degree at most $d$ and radius at most $t$. For a graph $G = (V, E)$, integers $d$ and $t$, and a vertex $v \in V$, let $B_G(v, d, t)$ be the subgraph rooted at $v$ that is induced by all vertices of $G[d]$ that are at distance $t$ or smaller from $v$. $B_G(v, d, t)$ is called a $(d, t)$-disk around $v$. From the definitions, the number of possible non-isomorphic $(d, t)$-disks is at most $N(d, t)$.

Proof of Lemma 4.4 The global algorithm of Theorem 2.7 can be easily simulated locally. To find $P(v)$, if $d(v) > \delta$, then the algorithm outputs $P(v) := \{v\}$. Otherwise, if the algorithm finds a vertex $u$ with $d(u) > \delta$ in the process of the local search, $u$ is ignored (the algorithm never check the neighbors of $u$). Thus the algorithm can behave like on the bounded-degree model. For any vertex $v$, $|P(v)| \leq 2^d = O(\delta^2/\epsilon)$. Each $u \in B_G(v, \delta, \mathcal{P})$ may be included in $P(w)$ of $w \in B_G(u, \delta, \mathcal{P})$. Then the algorithm checks at most all vertices in $B_G(v, \delta, 2\mathcal{P}) = B_G(v, \delta, O(\delta^2/\epsilon))$, and thus the query complexity for one call of $P(v)$ is at most $O(\delta^2/\epsilon)$.

\[\square\]

4.2 Abstract of the algorithm

How to construct a testing algorithm based on the partitioning oracle of Lemma 4.2 is almost the same with the way used in [17]. We use a distribution vector, which will be defined in Definition 4.3 of rooted subgraphs consisting of at most a constant number of vertices.

Definitions 4.3 For a graph $G = (V, E)$ and integers $d$ and $t$, let $\text{disk}_G(d, t)$ be the distribution vector of all $(d, t)$-disks of $G$, i.e., $\text{disk}_G(d, t)$ is a vector of dimension $N(d, t)$. Each entry of $\text{disk}_G(d, t)$ corresponds to some fixed rooted graph $H$, and counts the number of $(d, t)$-disks of $G[d]$ that are isomorphic to $H$. Note that $G[d]$ has $|V|$ different disks, thus the sum of entries in $\text{disk}_G(d, t)$ is $n$. Let $\text{freq}_G(d, t)$ be the normalized distribution, namely $\text{freq}_G(d, t) = \text{disk}_G(d, t)/n$.

For a vector $v = (v_1, \ldots, v_r)$, it’s $l_1$-norm is $||v||_1 = \sum_{i=1}^r |v_i|$. $l_1$-norm is also said the length of the vector. We say that two unit-length vectors $v$ and $u$ are $\epsilon$-close for $\epsilon > 0$ if $||v - u||_1 \leq \epsilon$.

By using a discussion that is the same with Theorem 3.1 in [17], the following lemma is proved.

Lemma 4.4 There exists functions $\lambda_{\mathbb{HFS}} = \lambda_{\mathbb{HFS}}(\mathcal{HFS}, \epsilon)$, $\lambda_{\mathbb{HFS}} = \lambda_{\mathbb{HFS}}(\mathcal{HFS}, \epsilon)$, $\lambda_{\mathbb{HFS}} = \lambda_{\mathbb{HFS}}(\mathcal{HFS}, \epsilon)$, $N_{\mathbb{HFS}} = N_{\mathbb{HFS}}(\mathcal{HFS}, \epsilon)$ such that for every $\epsilon > 0$ the following holds: For every $G_1, G_2 \in \mathcal{HFS}$ on $n \geq N_{\mathbb{HFS}}$ vertices, if $|\text{freq}_{G_1}(\mathcal{HFS}) - \text{freq}_{G_2}(\mathcal{HFS})| \leq \lambda_{\mathbb{HFS}}$, then $G_1$ and $G_2$ are $\epsilon$-close. □

A sketch of the algorithm is as follows. Let $G = (V, E)$ be a given graph and $P$ is a property to test. first we select some (constant) number $\ell = \ell(\epsilon)$ of vertices $v_i \in V$ ($i = 1, \ldots, \ell$) and find $P(v_i)$ given by the Theorem 2.7. This is done locally (shown by Lemma 1.2). Consider a graph $G' := P(v_1) \cup \cdots \cup P(v_\ell)$. Here, $\text{freq}_{G'}(d, t)$ and $\text{freq}_{G}(d, t)$ are very close in high provability. Next we calculate $\text{min}_{G \in P} |\text{freq}_{G'}(d, t) - \text{freq}_{G}(d, t)|$ approximately. There is a problem that the number of graphs in $P$ is generally infinite. However, to approximate it in a small error is enough for our object, and Thus it is enough to compare $G'$ with a constant number of vectors of $\text{freq}(d, t)$. (Note that calculating such a set of freq vectors requires much time. However, we can
say that there exists such a set. This means that the existence of the algorithm is assured.) The algorithm accepts $G$ if the approximate distance of $\min_{G \in P} |\text{freq}_{G'}(d, t) - \text{freq}_{G}(d, t)|$ is enough small, and rejects it otherwise.

The above algorithm is the same with the algorithm presented in [17] except two points: in our model, (1) $G$ is not a bounded-degree graph, and (2) $G$ is a multigraph. However, these differences is trivial. For the first difference, it is enough to add an ignoring-large-degree-vertex process, i.e., if the algorithm find a vertex $v$ having a degree larger than $d_{4.4}^t$, all edges incident to $v$ are ignored. By adding the above process, $G$ is regarded as $G|_d^t$. This modification does not effect the result by Lemma 3.1 For the second difference, the algorithm treats bounded-degree graphs as mentioned above, and the number of non-isomorphic multigraphs with $n$ vertices and degree upper bound $d_{4.4}$ is finite (bounded by $O(d_{4.4}^{n^2})$).

Proof of Theorem 2.8 Obtained from the above discussion.

5 Summary and future work

We introduced a natural class of multigraphs $\mathcal{HSF}$ representing scale-free networks, and we show that very wide subclass of it is hyperfinite (Theorem 2.7). By using this result, a very useful result that every property is testable on the class (Theorem 2.8) is obtained.

$\mathcal{HSF}$ is a class of multigraphs based on the hierarchical structure of isolated cliques. We may relax “isolated cliques” to “isolated dense subgraphs [13]” and may introduce a wider class. We considers that such classes also be hyperfinite. Finding such classes and proving their hyperfiniteness is important future work.

Acknowledgement

We are grateful for Assistant Professor Yuichi Yoshida of the National Institute of Informatics for his valuable suggestions. And we also appreciate Professor Osamu Watanabe of Tokyo Institute of Technology and Associate Professor Yushi Uno of Osaka Prefecture University for their fruitful discussions. We also thank “Algorithms on Big Data” project (ABD14) of CREST, JST, the ELC project (MEXT KAKENHI Grant Number 24106003), and JSPS KAKENHI Grant Number 24650006 through which this work was partially supported.

References

[1] R. Albert and A. -L. Barabási: Statistical mechanics of complex networks, Review of Modern Physics, Vol. 74, 2002, pp. 47–97.

[2] N. Alon, E. Fischer, I. Newman, and A. Shapira: A combinatorial characterization of the testable graph properties: it’s all about regularity, SIAM J. Comput., Vol. 39, No. 1, 2009, pp. 143–167.

[3] I. Benjamini, O. Schramm, and A. Shapira: Every minor-closed property of sparse graphs is testable, Proc. STOC 2008, ACM, 2008, pp. 393–402.

[4] A. Bottreau, Y. Métivier: Some remarks on the Kronecker product of graphs, Information Processing Letters, Vol. 68, Elsevier, 1998, pp. 55–61.
[5] A.Z. Broder, S.R. Kumar, F. Maghoul, R. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins and J.L. Wiener: Graph structure in the Web, Computer Networks, Vol. 33, 2000, pp.309–320.

[6] C. Cooper and R. Uehara: Scale free properties of random $k$-trees; Mathematics in Computer Science, Vol. 3, 2010, pp. 489–496.

[7] G. Elek: $L^2$-spectral invariants and convergent sequence of finite graphs, Journal of Functional Analysis, Vol. 254, No. 10, 2008, pp. 2667–2689.

[8] Y. Gao: The degree distribution of random $k$-trees, Theoretical Computer Science, Vol. 410, 2009, pp. 688–695.

[9] O. Goldreich (Ed.): Property Testing — Current Research and Surveys, LNSC 6390, 2010.

[10] O. Goldreich and D. Ron: Property testing in bounded degree graphs: Proc. STOC 1997, 1997, pp. 406–415.

[11] O. Goldreich, S. Goldwasser, and D. Ron: Property testing and its connection to learning and approximation: Journal of the ACM, Vol. 45, No. 4, July, 1998, pp. 653–750.

[12] A. Hassidim, J. A. Kelner, H. N. Nguyen, and K. Onak: Local graph partitions for approximation and testing, Proc. FOCS 2009, IEEE, pp. 22–31.

[13] H. Ito and K. Iwama, Enumeration of isolated cliques and pseudo-cliques, ACM Transactions on Algorithms, Vol. 5, Issue 4, Oct. 2009, Article 40.

[14] H. Ito and K. Iwama, and T. Osumi: Linear-time enumeration of isolated cliques, Proc. ESA2005, LNCS, 3669, Springer, 2005, pp. 119–130.

[15] J. Kleinberg and S. Lawrence: The structure of the Web, Science, Vol. 294, 2001, pp. 1894–1895.

[16] M. Mahdian and Y. Xu: Stochastic Kronecker graphs, Proc. WAW 2007, LNCS, 4863, Springer, 2007, pp. 179–186.

[17] I. Newman and C. Sohler: Every property of hyperfinite graphs is testable, Proc. STOC 2011, ACM, 2011, pp. 675–784.

[18] M. E. J. Newman: The structure and function of complex networks, SIAM Review, Vol. 45, 2003, pp. 167–256.

[19] R. Levi and D. Ron: A quasi-polynomial time partition oracle for graphs with an excluded minor, Proc. ICALP 2013 (1), LNCS, 7965, Springer, 2013, pp. 709–720.

[20] T. Shigezumi, Y. Uno, and O. Watanabe: A new model for a scale-free hierarchical structure of isolated cliques, Journal of Graph Algorithms and Applications, Vol. 15, No. 5, 2011, pp. 661–682.

[21] D.J. Watts and S.H. Strogatz: Collective dynamics of ‘small-world’ networks, Nature, Vol. 393, 1998, pp. 440–442.

[22] Z. Zhang, L. Rong, and F. Comellas: High-dimensional random apollonian networks, Physica A: Statistical Mechanics and its Applications, Vol. 364, 2006, 610–618.