Constant-time algorithms for complex networks

Hiro Ito
School of Informatics and Engineering,
The University of Electro-Communications, Tokyo, Japan
and
CREST, JST, Tokyo, Japan
Email: itohiro@uec.ac.jp

***Abstract*—In this paper, we study constant-time algorithms on complex networks. In a recent work presented in ESA 2016, the author introduced a natural class of multigraphs called hierarchical-scale-free (HSF) multigraphs and showed that a very wide subclass of HSF is hyperfinite. Based on this result, a surprising result such that every property is constant-time testable for the subclass was obtained. However, this result is theoretical, and the algorithm obtained directly from it may not be practical.**

In this paper, first we present the result of our paper of ESA 2016, and next we consider what kind of properties are "efficiently" testable in practice. For this purpose, we consider hereditary properties. We show that for hyperfinite graph class, e.g., HSF, any hereditary property can be represented by the forbidden graph subset such that the number of graphs in the subset is bounded by a constant. From this result, it can be considered that to test a hereditary property is relatively easy.

1. Introduction

1.1. Purpose of this paper

This paper studies how to solve problems on complex networks in very short time. For this purpose, the author recently presented a class of multigraphs called Hierarchical Scale Free, or HSF in short [14], [15], which may be able to model complex networks since it satisfies some typical properties, e.g. scale-freeness. For this class we showed the following result:

> For a very large subclass of HSF, every property can be testable in constant-time [15].

The purpose of this paper is

(1) introducing the above result, and
(2) considering the direction of research after the above result, and showing that hereditary properties have good characterizations to be efficiently testable.

1.2. Big data and property testing

How to handle big data is a very important issue in computer science. In the theoretical area, developing efficient algorithms for handling big data is an urgent task. For this purpose, constant-time algorithms look like they could be powerful tools, as they are able to read very small parts (constant size) of inputs.

Property testing is the most well-studied area in constant-time algorithms. A testing algorithm (or a tester) for a property accepts an input if it has the stipulated property and rejects it if it is far away from having the stipulated property with a 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 [10].

Property testing of graph properties has been well studied and many fruitful results have been obtained [2], [3], [7], [10], [11], [12], [13], [19], [21], [23], [24]. Testers on the graphs are separated into three groups according to model: 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, the characteristics of testable properties have been obtained [2]. However, graphs based on actual networks are usually sparse and thus unfortunately the dense-graph model does not fit. Studies on the bounded-degree model have been proceeding recently. One of the most important findings for this model is that every minor-closed property is testable [3]. This result can be extended to the surprising result that every property of a hyperfinite graph is testable [24]. (The definition of "hyperfinite" will be shown later.) However, graphs based on actual models have no degree bounds, i.e., it is known that web-graphs have hubs [1], [18], which have a large degree, and, unfortunately once again, these algorithms do not work for them.

1.3. A class HSF and constant-time testablity of the class

Consequently from the situation mentioned above, we need a new algorithm for treating actual big graphs. Typical big-data graph models are scale-free networks, which are characterized by the power-law degree distribution. Many models have been proposed for scale-free networks [1], [4], [5], [6], [9], [18], [22], [25], [26], [27], [28].

Recently, a promising model based on another property of a hierarchical isomorphic structure has been presented: If we look at a graph in a broad perspective, we find a

978-1-5090-5753-5/16 $31.00 © 2016 IEEE
DOI 10.1109/APWC.on.CSE.2016.12

similar structure to local structures. Shigezumi, Uno, and Watanabe [26] presented a model that is based on the idea of the hierarchical isomorphic structure of power-law distribution of isolated cliques. An idea of isolated cliques was given by Ito and Iwama [16], [17], and the definition is as follows. A *clique* is a subgraph in which there exists an edge between every pair of vertices. For a nonnegative integer $c \geq 0$, a *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 simply called an *isolated clique*.

Based on the model of [26], we introduced a class of multigraphs, hierarchical scale-free multigraphs (HSF, Definitions 3.2), which represents natural scale-free networks, and we showed the following result (Theorem 3.4):

> *Every property is testable on HSF if the power-law exponent*[1] *is greater than two* [15].

Given this result, many problems on actual scale-free big networks will prove to be solvable in constant time. Although this result is an application of the algorithms of [24], which is a result on bounded-degree graphs, HSF is not a class of bounded-degree graphs. This is the first universal result of constant-time testability on a class of graphs made by a model of scale-free networks.

1.4. And then what should we solve?

The above result (Theorem 3.4) is not perfect of course: There are some important problems that should be solved. Especially we mainly consider the following problem:

> The results is purely theoretical. In other words, some algorithms directly obtained from the result may require very large constant-time, and thus it may be completely useless in practice. However some properties may be testable in small constant-time, i.e., efficiently testable. We should try to clarify what properties are testable efficiently.

We will explore ways to get the solution on this problem. For this purpose, we will prove some lemmas on hereditary properties (properties that is closed under vertex deletion). From them, we observe that these properties look efficiently testable.

2. Preliminaries

2.1. Basic terms

In this paper, we consider undirected multigraphs without self-loops. We simply call this type of multigraph a "graph" in this paper and use $G = (V, E)$ to denote it, where V is the vertex set and E is the edge (multi)set. Sometimes V and E are denoted by $V[G]$ and $E[G]$, respectively. Henceforth, we use "set" to refer to a multiset for notational simplicity. Throughout this paper, n is used to denote the number of vertices of a graph, i.e., $|V| = n$.

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., $E_G(X, Y) = \{(x, y) \in E \mid x \in X, y \in Y\}$. $E_G(X, V\backslash X)$ is also simply written as $E_G(X)$. $|E_G(X)|$ is denoted by $d_G(X)$. For a vertex $v \in V$, the number of edges incident to v is called the *degree* of v. A singleton set $\{x\}$ is often written as x for notational simplicity. E.g., the degree of v is represented by $d_G(v)$. The subscript G in the above $E_G(*)$, $d_G(*)$, etc., may be omitted if it is clear.

For a vertex $v \in V$, $\Gamma_G(v)$ denotes the set of vertices adjacent to v, i.e., $\Gamma_G(v) := \{u \in V \mid (v, u) \in E\}$. Note that $|\Gamma_G(v)|$ may not be equal to $d_G(v)$ as parallel edges may exist. 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 $X \subseteq V$, a *contraction* of X is defined as an operation to (i) replace X with a new vertex v_X, (ii) replace each edge (v, u) in $E(X)$ ($v \in X, u \in V\backslash X$) with a new edge (v_X, u), and (iii) remove all edges between vertices in X. That is, by contracting $X \subseteq V$, a graph $G = (V, E)$ is changed to $G' = (V', E')$ such that

$$\begin{aligned} V' &= V\backslash X \cup \{v_X\}, \text{ and} \\ E' &= E\backslash\{(v, u) \mid v \in X, u \in V\} \\ &\quad \cup\{(v_X, u) \mid (v, u) \in E, v \in X, u \in V - X\}. \end{aligned}$$

We identify the above $(v_X, u) \in E'$ with $(v, u) \in E$. In other words, we say that (v, u) remains in G' (as (v_X, u)). Note that the graphs are multigraphs, and thus if there are two edges $(v, u), (v', u) \in E$ for $v, v' \in X$, $v \neq v'$ and $u \in V\backslash X$, then two parallel edges, both represented by (v_X, u), one of which corresponds to (v, u) and the other of which corresponds to (v', u), are added to E'. Also note that none of the graphs considered in this paper contain self-loops, and hence an edge $(v, v') \in E$ with $v, v' \in X$ is removed by contracting X.

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, for short) is a (possibly infinite) family of graphs, which is closed under isomorphism.

2.2. Testers and isolated cliques

Definitions 2.1 (distance, ϵ-far, and ϵ-close). Let $G = (V, E)$ and $G' = (V', E')$ be two graphs with $|V| = |V'| = n$ vertices. Let $m(G, G')$ be the number of edges that need 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[2] $\mathrm{dist}(G, G') = m(G, G')/n$. We say that

1. This is a parameter of HSF. For the definition, see the sentence just after Definitions 3.1.

2. The distance defined here may be larger than 1 as $m(G, G') > n$ may occur. (In the bounded-degree model it is defined as $\mathrm{dist}(G, G') = m(G, G')/dn$.) However, here we consider sparse graphs and they have an implicit upper bound of the average (not possibly maximum) degree, say d, and thus $\mathrm{dist}(G, G')$ is bounded by d.

G and G' are ϵ-*far* if $\text{dist}(G, G) > \epsilon$; otherwise ϵ-*close*. 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 ϵ-*far* from P if $\text{dist}(G, P) > \epsilon$; otherwise ϵ-*close*.

Definitions 2.2 (testers). A *testing algorithm* for a property P is an algorithm that, given query access to a graph G, accepts every graph from P with a probability of at least $2/3$, and rejects every graph that is ϵ-far from P with probability at least $2/3$. Oracles in the general graph model are: for any vertex v, the algorithm may ask for the degree $d(v)$, and may ask for the ith neighbor of the vertex (for $1 \le i \le d(v)$).[3] The number of queries made by an algorithm to the given oracle is called the *query complexity* of the algorithm. If the query complexity of a testing algorithm is a constant, independent of n (but it may depend on ϵ), then the algorithm is called a *tester*[4]. A (graph) property is *testable* if there is a tester for the property.

Definitions 2.3 (isolated cliques [16]). For a graph $G = (V, E)$ and a real number $c \ge 0$, a vertex subset $Q \subseteq V$ is called a c-*isolated clique* if Q is a clique (i.e., $(u, v) \in E$, for all $u, v \in Q$ and $u \ne v$) and $d_G(Q) < c|Q|$. A 1-isolated clique is sometimes called an *isolated clique*. In this paper, we don't use $c > 1$ except section **6** (summary and future work).

Definitions 2.4. Let $\mathcal{E}(G)$ be the graph obtained from G by contracting all isolated cliques. Two distinct isolated cliques never overlap, except in the special case of *double-isolated-cliques*, which consists of two isolated cliques with size k sharing $k - 1$ vertices. A double-isolated-clique Q has no edge between Q and the other part of the graph (i.e., $d_G(Q) = 0$), and thus we specially define that a double-isolated-clique in G is contracted into a vertex in $\mathcal{E}(G)$. Under this assumption, $\mathcal{E}(G)$ is uniquely defined.

2.3. Hyperfiniteness and the universal results of Newman and Sohler [24]

The following result is one of the most important results in the area of property testing. This result is applicable for the class of hyperfinite bounded-degree graphs.

Definitions 2.5 (hyperfinite [8]). For real numbers $t > 0$ and $\epsilon > 0$, a graph $G = (V, E)$ consisting of n vertices is (t, ϵ)-*hyperfinite* if one can remove at most ϵn edges from G and obtain a graph whose connected components have size at most t. For a function $\rho : \boldsymbol{R}^+ \to \boldsymbol{R}^+$, G is ρ-*hyperfinite* if it is $(\rho(\epsilon), \epsilon)$-hyperfinite for all $\epsilon > 0$. A family $\mathcal{G}$ of graphs is ρ-*hyperfinite* if all $G \in \mathcal{G}$ are ρ-hyperfinite. A family $\mathcal{G}$ of graphs is *hyperfinite* if there exists a function ρ such that $\mathcal{G}$ is ρ-hyperfinite.

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

Theorem 2.6 ([24]). 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 in that the result works on bounded-degree graphs and it is natural to consider that actual scale-free networks do not have a degree bound.

3. Classes SF and HSF, and Hyperfiniteness

3.1. Class Scale-Free (SF)

In [15], we applied the universal algorithm of [24] to scale-free networks. We formalized two natural classes, $\mathcal{SF}$ and $\mathcal{HSF}$ that represent scale-free networks[5]. The latter is a subclass of the former.

Definitions 3.1 (Scale-Free Graphs [15]). For positive real numbers $c > 1$ and $\gamma > 1$, a class of *scale-free graphs (SF)* $\mathcal{SF}(c, \gamma)$ consists of (multi)graphs $G = (V, E)$ for which the following condition holds: Let ν_i be the number of vertices v with $d(v) = i$. Then:

$$\nu_i \le cni^{-\gamma}, \quad \forall i \in \{2, 3, \ldots, \}. \tag{1}$$

□

The above property (1) is generally called a power-law and we call γ a *power-law exponent*. In many actual scale-free networks, it is said that $2 < \gamma < 3$ [1]. That is, $\mathcal{SF}$ is a class of multigraphs that obey the power-law degree distribution.

It was shown that that this class is ϵ-close to a bounded-degree class if $\gamma > 2$ (Lemma 3.5). Moreover in [15], after showing this property, the hyperfiniteness of the class was shown.

3.2. Cluster coefficient and hyperfiniteness

Hyperfiniteness seems to be closely related to a high cluster coefficient, where the cluster coefficient $\text{cl}(G)$ of a graph $G = (V, E)$ is defined as follows[6]: For a graph $G = (V, E)$ and a vertex $v \in V$, the (local) *cluster coefficient* of v is

$$\text{cl}_G(v) := \frac{|\{(u, w) \in E \mid u, w \in \Gamma_G(v), u \ne w\}|}{\binom{|\Gamma_G(v)|}{2}}.$$

3. Although asking 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.

4. In this paper, a tester may be nonuniform, i.e., it may depend on n and ϵ.

5. $\mathcal{HSF}$ was introduced in the preliminary version of this paper [14]. However, the definition in this paper is more general (wider) than in the preliminary version.

6. There is another way to define the cluster coefficient: $3 \times$ (# of cycles of length three)/(# of paths of length two). Although these two values are different generally, they are close under the assumption of the power-law degree distribution.

And the *cluster coefficient* of G is

$$\mathrm{cl}(G) := \frac{1}{n}\sum_{v\in V} \mathrm{cl}_G(v).$$

It is said that $\mathrm{cl}(G)$ is $\Theta(1)$ for actual social networks, while $\lim_{n\to\infty} \mathrm{cl}(G) = 0$ for random graphs.

These three characterizations, "high clustering coefficient," "existence of isolated cliques," and "hyperfiniteness" appear to be closely related to each other. In fact, it is readily observed that if $\mathrm{cl}_G(v) = 1$ for a bounded-degree graph G (the degree bound is d), then G consists of only (completely) isolated cliques with size at most $d+1$, and G is $(d+1, 0)$-hyperfinite!

Unfortunately, however, it is also observed that for any $0 < c < 1$, there is a class of bounded-degree graphs G such that $\lim_{n\to\infty} \mathrm{cl}(G) = c$ and it is not (t, ϵ)-hyperfinite for any pair of constants t and $\epsilon < 1/2$, e.g., $G = (V, E)$ consists of n/d cliques of size d, and random $n/2$ edges between vertices in different cliques. Each vertex has $d-1$ adjacent vertices in its clique and one adjacent vertex outside the clique. See Fig. 1. To separate this graph into constant-sized connected components, almost all of the edges between cliques (their number is $n/2$) must be removed. It follows that this graph is not hyperfinite for any $\epsilon < 1/2$. On the other hand, the cluster coefficient is

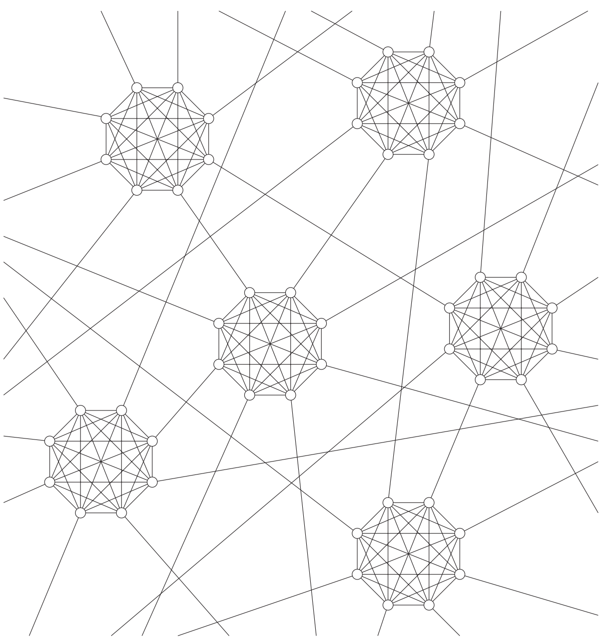

Figure 1. An example of non-hyperfinite graphs having high cluster coefficient.

$$\mathrm{cl}(G) = \frac{\binom{d-1}{2}}{\binom{d}{2}} = 1 - \frac{2}{d} > c$$

if $d > \frac{2}{1-c}$, i.e., the cluster coefficient of graphs in this class are greater than c.

3.3. Hierarchical structure of complex networks and class HSF

However, we do not need to give up here, as the above model is very special, e.g., by contracting each isolated clique, it becomes a mere random graph with n/d vertices[7]. From this fact, the hierarchical structure of a high cluster coefficient looks important. The model presented by [26] has such a structure. Based on this model, we presented the following class of multigraphs:

Definitions 3.2 (Hierarchical Scale-Free Graphs [15]). For positive real numbers $c, \gamma > 1$ and a positive integer $n_0 \geq 1$, a class of *hierarchical scale-free graphs (HSF)* $\mathcal{HSF} = \mathcal{HSF}(c, \gamma, n_0)$ consists of (multi)graphs $G = (V, E)$ for which the following conditions hold:

(i) $G \in \mathcal{SF}(c, \gamma)$
(ii) Consider the infinite sequence of graphs $G_0 = G$, $G_1 = \mathcal{E}(G_0)$, $G_2 = \mathcal{E}(G_1)$, …. If $|V[G_i]| \geq n_0$, then G_i includes at least one isolated clique $Q \subseteq V$ with $|Q| \geq 2$. (Note that if G_k has no such isolated clique, then $G_k = G_{k+1} = G_{k+2} = \cdots$.)

In [15] the following results were shown.

Theorem 3.3 ([15]). For any $\mathcal{HSF} = \mathcal{HSF}(c, \gamma, n_0)$ with $\gamma > 2$ and any real number $\epsilon > 0$, there is a real number $t_{3.3} = t_{3.3}(\mathcal{HSF}, \epsilon)$ such that $\mathcal{HSF}$ is $(t_{3.3}, \epsilon)$-hyperfinite.

In [15], a global algorithm for obtaining the partition realizing the hyperfiniteness of Theorem 3.3 was given. The algorithm is deterministic, i.e., if a graph and the parameter ϵ are fixed, then the partition is also fixed. The algorithm can be easily revised to a local algorithm and we obtain a deterministic partitioning oracle to get the partition (Lamma 4.2). Note that almost all algorithms for partitioning oracles presented to date have been randomized algorithms[8]. By using this partitioning oracle and an argument similar to one used in [24], the following theorem follows.

Theorem 3.4 ([15]). Any property is testable for $\mathcal{HSF}(c, \gamma, n_0)$ with $\gamma > 2$.

As stated earlier, for the bounded-degree model, Newman and Sohler [24] presented a universal tester (which can test any property) for hyperfinite graphs. In the general graph model, although some works have tried to found universal tester [7], [19], [23], these results are weaker than for the bounded-degree graph model and the dense graph model.

This paper gives a universal tester that can test every property on a natural class of scale-free multigraphs in constant time. This is the first result for universal constant-time algorithms which cover a class of graphs made by a model of scale-free networks.

7. However, note that this model is not useless, since it is investigated in some works [20].

8. The algorithm for testing forests presented by Kusumi and Yoshida [19] may be only deterministic one so far. That is, our partitioning oracle looks the first deterministic one for a graph class that includes cycles.

In this paper we omit to show the proofs of Theorems 3.3 and 3.4 (see [15]). However we only show that $\mathcal{SF}$ (and thus $\mathcal{HSF}$ also) can be treated as a bounded-degree graph for testing in the next subsection.

3.4. Degree bounding

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

Lemma 3.5. For any $\mathcal{SF} = \mathcal{SF}(c, \gamma)$ with $\gamma > 2$, and any positive real number $\epsilon > 0$, there is a constant $\delta_{3.5} = \delta_{3.5}(\epsilon, c, \gamma)$ such that for any graph $G \in \mathcal{SF}$, $G|\delta_{3.5}$ is ϵ-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 converge to a constant ($\zeta(\gamma) < 1+(\gamma-1)^{-1}$) for any $\gamma > 1$. We introduce a generalization of this function by using a positive integer $k \geq 1$ as $\zeta(k, \gamma) = \sum_{i=k}^{\infty} i^{-\gamma}$. Note that $\zeta(\gamma) = \zeta(1, \gamma)$.

Lemma 3.6. For any $\epsilon > 0$ and $\gamma > 1$, there is an integer $k_{3.6} = k_{3.6}(\epsilon, \gamma) \geq 1$ such that $\zeta(k_{3.6}, \gamma) < \epsilon$.

Proof: It is clear from the above fact that $\zeta(\gamma)$ converges for every $\gamma > 1$. □

Proof of Lemma 3.5: Let d be an arbitrary positive integer. Let m_d be the number of removed edges to make $G|d$ from G. From (1),

$$m_d = \sum_{i=d+1}^{\infty} i\nu_i \leq \sum_{i=d+1}^{\infty} cni^{-(\gamma-1)} = cn\zeta(d+1, \gamma-1).$$

From the assumption of $\gamma > 2$ and Lemma 3.6, $\zeta(d + 1, \gamma - 1) < \epsilon/c$ if $d + 1 \geq k_{3.6}(\epsilon/c, \gamma - 1)$. Thus by letting $\delta_{3.5}(\epsilon, c, \gamma) = k_{3.6}(\epsilon/c, \gamma - 1) - 1$, we have $m_{\delta_{3.5}} < \epsilon n$. □

From here, we denote the above $\delta_{3.5}(\epsilon, c, \gamma)$ by δ for notational simplicity.

4. Testing Algorithm

4.1. Deterministic partitioning oracle

The global partitioning algorithm, which is presented in [15], of Theorem 3.3 can be easily revised to run locally, i.e., a "partitioning oracle" based on this algorithm can be obtained. A partitioning oracle, which calculates a partition realizing hyperfiniteness locally, was introduced by Benjamini, et al. [3] implicitly and by Hassidim, et al. [13] explicitly. It is a powerful tool for constructing constant-time algorithms for sparse graphs. It has been revised by some researchers and Levi and Ron's algorithm [21] is the fastest to date. As mentioned before almost all algorithms for partitioning oracles presented to date have been randomized algorithms. Our algorithm, however, does not use any random valuable and it runs deterministically. That is, we call it a *deterministic partitioning oracle*, which is rigorously defined as follows[9]:

Definitions 4.1. $\mathcal{O}$ is a deterministic (t, ϵ)-partitioning oracle for a class of graphs $\mathcal{C}$, if, given query access to a graph $G = (V, E)$, it provides query access to a partition $\mathcal{P}$ of G. For a query about $v \in V$, $\mathcal{O}$ returns $\mathcal{P}(v)$. The partition has the following properties: (i) $\mathcal{P}$ is a function of G, t, and ϵ. (It does not depend on the order of queries to $\mathcal{O}$.) (ii) For every $v \in V$, $|\mathcal{P}(v)| \leq t$ and $\mathcal{P}(v)$ induces a connected subgraph of G. (iii) If $G \in \mathcal{C}$, then $|\{(u, v) \in E \mid \mathcal{P}(u) \neq \mathcal{P}(v)\}| \leq \epsilon|V|$.

Lemma 4.2 ([15]). There is a deterministic $(t_{3.3}, \epsilon)$-partitioning oracle $\mathcal{O}_{\mathcal{HSF}}$ for $\mathcal{HSF}$ with $\gamma > 2$ with query complexity $\delta^{O(\delta^2/\epsilon+n_0)}$ for one query.

We omit to show the proof of this lemma (see [15]).

4.2. (d, t)-disks and frequency vectors

The method of constructing a testing algorithm based on the partitioning oracle of Lemma 4.2 is almost the same as the method used in [24]. Before showing the algorithm, we introduce some notation 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 a 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 a maximum degree of d and a maximum radius of 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 less from v. $B_G(v, d, t)$ is called a (d, t)-*disk* around v. From these definitions, the number of possible non-isomorphic (d, t)-disks is at most $N(d, t)$.

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

9. However, since Levi and Ron's algorithm [21] looks fast, using it may be better in practice.

By using the same discussion as in Theorem 3.1 in [24], the following lemma is proven.

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

4.3. Abstract of the algorithm

A sketch of the algorithm is as follows. Let $G = (V, E)$ be a given graph and P be 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 $\mathcal{P}(v_i)$ given by Theorem 3.3. This is done locally (shown by Lemma 4.2). Consider a graph $G' := \mathcal{P}(v_1) \cup \cdots \cup \mathcal{P}(v_\ell)$. Here, $\mathrm{freq}_G(d, t)$ and $\mathrm{freq}_{G'}(d, t)$ are very close with high probability.

Next, we calculate $\min_{G \in P} |\mathrm{freq}_{G'}(d, t) - \mathrm{freq}_G(d, t)|$ approximately. There is a problem in that the number of graphs in P is generally infinite. However, to approximate it with a small error is adequate for our objective, and thus

> it is sufficient to compare G' with a constant number of vectors of $\mathrm{freq}(d, t)$.

(Note that calculating such a set of frequency 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 $\min_{G \in P} |\mathrm{freq}_{G'}(d, t) - \mathrm{freq}_G(d, t)|$ is small enough, and otherwise it is rejected.

The above algorithm is the same as the algorithm presented in [24] except for two points: in our model: (1) G is not a bounded-degree graph, and (2) G is a multigraph. However, these differences are 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 $d_{4.4}$, all edges incident to v are ignored. By adding this process, G is regarded as $G|d_{4.4}$. This modification does not effect the result by Lemma 3.5. 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 3.4: Obtained from the above discussion. □

5. Practically tastable properties

Theorem 3.4 assures that every property is testable (in constant-time). However, this result is only theoretical one. In fact, an algorithm directly obtained from the proof of the theorem is generally impractical. Because, it requires the set of the frequency vectors $\mathrm{freq}(d, t)$ that satisfy the property, and the number of such vectors should be generally very huge[10]. Moreover, as pointed out in the proof, we only know the existence of the set of the vectors, and we don't know any efficient way to get it.

Here we try to find properties that is testable in not large constant-time. We are considering the following properties:

Definitions 5.1. A property is called *hereditary* if it's closed under vertex-deletions, i.e., for any $G \in P$, any induced subgraph G' of G is also in P.

Many well-studied properties are known to be hereditary, e.g., planar, k-colorable (for any integer $k \geq 0$), H-free (for any graph H), any minor-closed property (i.e., closed under vertex- and edge-deletions and edge-contractions), any monotone property (i.e., closed under vertex- and edge-deletions), perfect, etc.

The following useful result has been known:

Lemma 5.2. For any hereditary property P, there is a (possibly infinite) set $\mathcal{H}_P$ of graphs such that $G \in P$ if and only if G doesn't include any $H \in \mathcal{H}_P$ as an induced subgraph.

Proof: Let $\mathcal{H}_\mathcal{P}$ be the set of graphs H such that $H \notin P$. We show that $\mathcal{H}_\mathcal{P}$ is the desired set.

"If part" is clear since if $G \notin P$, then clearly $G \in \mathcal{H}_\mathcal{P}$ itself is a subset of G.

For showing "only-if part," we assume that G includes a graph $H \in \mathcal{H}_\mathcal{P}$ as an induced subgraph. From the definition of $\mathcal{H}_\mathcal{P}$, $H \notin P$. From that P is hereditary, $G \notin P$. (Since if $G \in P$, then an induced subgraph H of G must not be in P.) □

Lemma 5.3. Let P be a hereditary property such that $\forall H \in \mathcal{H}_P$ is connected. For a ρ-hyperfinite graph class $\mathcal{G}$ and a positive real vale $\epsilon > 0$, there are an integer $t = t_{5.3}(\rho, \epsilon)$ and a finite set of graphs $\mathcal{H}_{P,t}$ that includes graphs having at most t vertices such that

(i) for any $G \in P \cap \mathcal{G}$, G doesn't include any $H \in \mathcal{H}_{P,t}$ as an induced subgraph, and
(ii) for any $G \in \mathcal{G}$ that is ϵ-far from P, G includes an $H \in \mathcal{H}_{P,t}$ as an induced subgraph.

Proof: Let $t := \rho(\epsilon/2)$ and

$$\mathcal{H}_{P,t} := \{H \in \mathcal{H}_P \mid H \text{ has at most } t \text{ vertices}\}.$$

The number of non-isomorphic graphs with at most t vertices is at most $2^{\binom{t}{2}}$, and thus $\mathcal{H}_{P,t}$ is a finite set.

Assume that $G \in P$. From Lemma 5.2 and $\mathcal{H}_{P,t} \subseteq \mathcal{H}_P$, G contains no $H \in \mathcal{H}_{P,t}$ as an induced subgraph. Therefore (i) is proven.

Next, assume that G is ϵ-far from P. G is ρ-hyperfinite, and hence G is $(\rho(\epsilon/2), \epsilon/2)$-hyperfinite. Thus there is a graph G' such that every connected component in it consists of at most $t = \rho(\epsilon/2)$ vertices, and $\mathrm{dist}(G, G') \leq \epsilon/2$. From that G is ϵ-far from P, it follows that G' is $\epsilon/2$-far from P. Thus G' includes at least one $H \in \mathcal{H}_P$. From that $\forall H \in \mathcal{H}_P$

10. The size is in fact bounded by a constant. However it is really large constant for some ϵ.

is connected and that every connected component of G' has at most t vertices, it follows that G' includes at least one $H \in \mathcal{H}_{P,t}$, and thus G does also. Therefore (ii) is proven. □

From Lemma 5.3, we may have a practical algorithm as follows:

Procedure HEREDITARY-HYPERFINITE-TEST
begin

Step 1 select constant number s of vertices $v_1, \ldots, v_s$ uniformly at random;
Step 2 find connected components $C_1, \ldots, C_s$ such that $v_i \in C_i$ and C_i is the connected component obtained by the partitioning oracle;
Step 3 if at least one C_i includes an $H \in \mathcal{H}_{P,t}$ as an induced subgraph, then "reject," and otherwise "accept."

end.

The practicality of this algorithm depends on the size of $|P_{\mathcal{H}}, t|$ and t. For example, triangle-free, claw-free, and k-leaves-star-free (for small k) is considered to be easy.

6. Summary and future work

We first presented the result of [15] such that a natural class of multigraphs $\mathcal{HSF}$ representing scale-free networks, and that a wide subclass of it is hyperfinite (Theorem 3.3), and that every property is testable on the class (Theorem 3.4).

Next we considered that a property that may be tested in practice. We showed that hereditary properties look effectively testable.

For future work, for theoretical side, extending the results of Theorems 3.3 and 3.4 is important problem. HFS doesn't include every graphs model by other known complex networks. We are considering to extend the constant-time testable set of multigraphs to include such known models.

For practical side, by using real network data, to check how our tester works and what properties, especially hereditary ones, are testable efficiently is also an important future problem.

Acknowledgements

We are grateful to Associate Professor Yuichi Yoshida of the National Institute of Informatics for his valuable suggestions. We also appreciate the fruitful discussions with Professor Ilan Newman of University of Haifa, Professor Osamu Watanabe of the Tokyo Institute of Technology, and Associate Professor Yushi Uno of Osaka Prefecture University. We also would like to thank the anonymous referees for their variable comments. Finally, we thank the Algorithms on Big Data project (ABD14) of CREST, JST, the ELC project (MEXT KAKENHI Grant Number 24106003), and JSPS KAKENHI Grant Number 15K11985 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, P. Raghavan, S. Rajagoplalan, 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] A. Czumaj, M. Monemizadeh, K. Onak, and C. Sohler: Planar Graphs: Random Walks and Bipartiteness Testing, Proceedings of FOCS 2011, pp. 423–432.

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

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

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

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

[12] 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.

[13] 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.

[14] H. Ito, Every property is testable on a natural class of scale-free multigraphs, arXiv: 1504.00766, Cornell University, April 6, 2015.

[15] H. Ito, Every property is testable on a natural class of scale-free multigraphs, Proc. of ESA 2016, LIPICS, Vol. 49 (ISBN 978-3-95977-005-7), 2016, pp. 21:2–21:15.

[16] H. Ito and K. Iwama, Enumeration of isolated cliques and pseudo-cliques, ACM Transactions on Algorithms, Vol. 5, Issue 4, Oct. 2009, Article 40 (pp. 1–13).

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

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

[19] M. Kusumoto and Y. Yoshida: Testing forest-isomorphizm in the adjacency list model, Proc. of ICALP2014 (1), LNSC 8572, 2014, pp. 763–774.

[20] A. Lancichinetti, S. Fortunato, and F. Radicchi: Benchmark graphs for testing community detection algorithms, arXiv: 0805.4770v4, Cornell University, Oct. 3, 2008.

[21] 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. (Journal version: R. Levi and D. Ron: A quasi-polynomial time partition oracle for graphs with an excluded minor, ACM Transactions on Algorithms, Vol. 11, No. 3, 2014, Article 24 (pp. 1–13).)

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

[23] S. Marko, D. Ron: Approximating the distance to properties in bounded-degree and general sparse graphs., ACM Transactions on Algorithms, Vol. 5, Issue. 2, 2009, Article No. 22.

[24] I. Newman and C. Sohler: Every property of hyperfinite graphs is testable, Proc. STOC 2011, ACM, 2011, pp. 675–784. (Journal version: I. Newman and C. Sohler: Every property of hyperfinite graphs is testable, SIAM J. Comput., Vol. 42, No. 3, 2013, pp. 1095–1112.)

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

[26] 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.

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

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