Every Property Is Testable on a Natural Class of Scale-Free Multigraphs*

Hiro Ito

School of Informatics and Engineering, The University of Electro-Communications (UEC)/Tokyo, Japan; and
CREST, JST/Tokyo, Japan
itohiro@uec.ac.jp

Abstract

In this paper, we introduce a natural class of multigraphs called hierarchical-scale-free (HSF) multigraphs, and consider constant-time testability on the class. We show that a very wide subclass of HSF is hyperfinite. Based on this result, an algorithm for a deterministic partitioning oracle can be constructed. We conclude by showing that every property is constant-time testable on the above subclass of HSF. This algorithm utilizes findings by Newman and Sohler of STOC'11. However, their algorithm is based on a bounded-degree model, while it is known that actual scale-free networks usually include hubs, which have a very large degree. HSF is based on scale-free properties and includes such hubs. This is the first universal result of constant-time testability on a class of graphs made by a model of scale-free networks, and it has the potential to be applicable on a very wide range of scale-free networks.

1998 ACM Subject Classification G.2.2 Graph Theory: Graph algorithms, G.3 Probability and Statistics: Probabilistic algorithms

Keywords and phrases constant-time algorithms, scale-free networks, complex networks, isolated cliques, hyperfinite

Digital Object Identifier 10.4230/LIPIcs.ESA.2016.51

1 Introduction

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, 18, 20, 22, 23]. 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,

* This work was partially supported by the Algorithms on Big Data project (ABD14) of CREST, JST, the ELC project (MEXT KAKENHI Grant Number 24106003), and JSPS KAKENHI Grant Numbers 24650006 and 15K11985.

24th Annual European Symposium on Algorithms (ESA 2016).
Editors: Piotr Sankowski and Christos Zaroliagis; Article No. 51; pp. 51:1–51:12
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 [23]. However, graphs based on actual models have no degree bounds, i.e., it is known that web-graphs have hubs [1, 17], which have a large degree, and, unfortunately once again, these algorithms do not work for them.

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, 17, 21, 24, 25, 26, 27]. 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 similar structure to local structures. Shigezumi, Uno, and Watanabe [25] 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 [15, 16], and the definition is as follows. 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 [25], we introduce a class of multigraphs, hierarchical scale-free multigraphs (HSF, Definitions 1.8)[1], which represents natural scale-free networks. We show the following result (Theorem 1.10):

Every property is testable on HSF if the power-law exponent[2] *is greater than two.*

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 [23], 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.

1.1 Definitions

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.

[1] In a preliminary version of this paper, [14], the definition of HSF is different. The definition in this paper is far more general (wider) than in the preliminary version.
[2] This is a parameter of HSF. For the definition, see the sentence just after Definitions 1.7.

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

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

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.

▶ **Definition 1.1** (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[3] $\text{dist}(G, G') = m(G, G')/n$. We say that G and G' are *ϵ-far* if $\text{dist}(G, G) > \epsilon$; otherwise *$\epsilon$-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 *$\epsilon$-close*.

▶ **Definition 1.2** (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 $1 \leq i \leq d(v)$).[4] 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*[5]. A (graph) property is *testable* if there is a tester for the property.

▶ **Definition 1.3** (isolated cliques [15])**.** 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., $(u, v) \in E$, for all $u, v \in Q$ and $u \neq v$) 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 **4** (summary and future work).

---

[3] 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 $\text{dist}(G, G') = m(G, G')/dn$.) However, here we consider sparse graphs and they have an implicit upper bound of the average (not possibly maximum) degree, say d, and thus $\text{dist}(G, G')$ is bounded by d.

[4] 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.

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

▶ **Definition 1.4.** 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.

▶ **Definition 1.5** (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 a bounded-degree model. From the viewpoint of testing, the importance of hyperfiniteness stems from the following result.

▶ **Theorem 1.6** ([23])**.** *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.

1.2 Our contribution and related work

In this paper, we apply the universal algorithm of [23] to scale-free networks. We formalize two natural classes, $\mathcal{SF}$ and $\mathcal{HSF}$ that represent scale-free networks[6]. The latter is a subclass of the former.

▶ **Definition 1.7.** 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:

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

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.

We show that this class is ϵ-close to a bounded-degree class if $\gamma > 2$ (Lemma 2.1).

After showing this property, we show the hyperfiniteness of the class. Hyperfiniteness seems to be closely related to a high clustering coefficient, where the cluster coefficient $\mathrm{cl}(G)$ of a graph $G = (V, E)$ is defined as[7]:

$$\mathrm{cl}(G) := \frac{1}{n} \sum_{v \in V} \mathrm{cl}_G(v), \qquad \mathrm{cl}_G(v) := \frac{|\{(u, w) \in E \mid u, w \in \Gamma_G(v), u \neq w\}|}{\binom{|\Gamma_G(v)|}{2}}.$$

[6] $\mathcal{HSF}$ 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.

[7] 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.

Sometimes $\mathrm{cl}_G(v)$ is called the *local cluster coefficient* of v. It is said that $\mathrm{cl}(G)$ is $\Theta(1)$ for many classes that model 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). To separate this graph into constant-sized connected components, almost all of the edges between cliques (their number is $n/2$) must be removed.

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[8]. From this fact, the hierarchical structure of a high cluster coefficient looks important. The model presented by [25] has such a structure. Based on this model, we present the following class of multigraphs:

▶ **Definition 1.8** (Hierarchical Scale-Free Graphs)**.** 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 $G_k = G_{k+1} = G_{k+2} = \cdots$.)

We show the following results.

▶ **Theorem 1.9.** *For any $\mathcal{HSF} = \mathcal{HSF}(c, \gamma, n_0)$ with $\gamma > 2$ and any real number $\epsilon > 0$, there is a real number $t_{1.9} = t_{1.9}(\mathcal{HSF}, \epsilon)$ such that $\mathcal{HSF}$ is $(t_{1.9}, \epsilon)$-hyperfinite.*

We give a global algorithm for obtaining the partition realizing the hyperfiniteness of Theorem 1.9. 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 3.2). Note that almost all algorithms for partitioning oracles presented to date have been randomized algorithms[9]. By using this partitioning oracle and an argument similar to one used in [23], we get the following main theorem.

▶ **Theorem 1.10.** *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 [23] 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, 18, 22], these results are weaker than for the bounded-degree graph model and the dense graph model.

[8] However, note that this model is not useless, since it is investigated in some works [19].

[9] The algorithm for testing forests presented by Kusumi and Yoshida [18] may be only deterministic one so far. That is, our partitioning oracle looks the first deterministic one for a graph class that includes cyclic graphs.

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.

2 Hyperfiniteness and a Global Partitioning Algorithm

2.1 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 2.1.** *For any $\mathcal{SF} = \mathcal{SF}(c, \gamma)$ with $\gamma > 2$, and any positive real number $\epsilon > 0$, there is a constant $\delta_{2.1} = \delta_{2.1}(\epsilon, c, \gamma)$ such that for any graph $G \in \mathcal{SF}$, $G|\delta_{2.1}$ 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 2.2.** *For any $\epsilon > 0$ and $\gamma > 1$, there is an integer $k_{2.2} = k_{2.2}(\epsilon, \gamma) \geq 1$ such that $\zeta(k_{2.2}, \gamma) < \epsilon$.*

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

Proof of Lemma 2.1. 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 2.2, $\zeta(d+1, \gamma-1) < \epsilon/c$ if $d+1 \geq k_{2.2}(\epsilon/c, \gamma-1)$. Thus by letting $\delta_{2.1}(\epsilon, c, \gamma) = k_{2.2}(\epsilon/c, \gamma-1) - 1$, we have $m_{\delta_{2.1}} < \epsilon n$. ◀

From here, we denote the above $\delta_{2.1}(\epsilon, c, \gamma)$ by δ for notational simplicity.

2.2 Hierarchical contraction, structure tree, and coloring

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 $\{W_1, \ldots, W_k\}$ is called a *partition* of V. Below, we explain a global algorithm for obtaining a partition of V realizing the hyperfiniteness of a graph in $\mathcal{HSF}$ with $\gamma > 2$, i.e., $|W_i|$ is bounded by a constant and the number of edges between different W_i and W_j is, at most, ϵn. First, we give a base algorithm.

procedure HIERARCHICALCONTRACTION(G)
begin
  $i := 0$, $G_0 := G$
  while there exists an isolated clique in $G_i = (V_i, E_i)$ **do**
    $i := i + 1$, $G_i := \mathcal{E}(G_{i-1})$
  enddo
end.

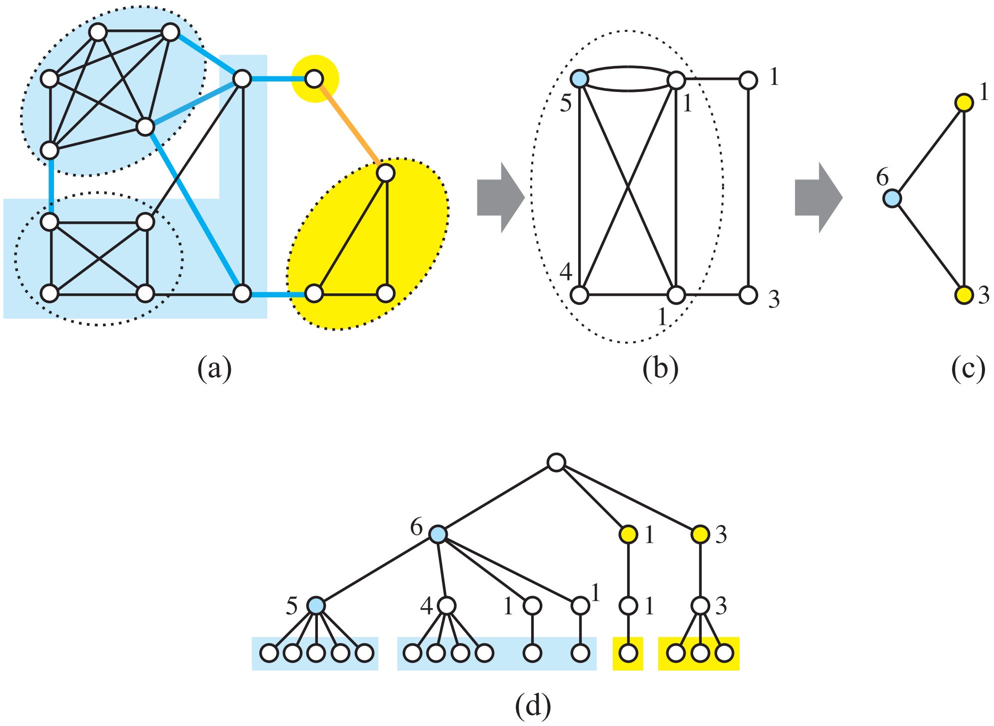

Figure 1 An example of HIERARCHICALCONTRACTION, the structure tree T, and the coloring: Here, we assume $\delta/\epsilon = 4.5$; the number beside a vertex is $w(*)$; the dotted circles are isolated cliques; colored areas are blue or yellow components.

We denote $G_i = (V_i, E_i)$ for $i \in \{0, 1, \ldots\}$. Let $G_k = (V_k, E_k)$ be the final graph of HIERARCHICALCONTRACTION(G). From the definitions of HSF, $|V_k| < n_0$. See Fig. 1 (a)–(c) for an example of applying this procedure.

The trail of the contraction can be represented by a rooted tree $T = (V[T], E[T])$, which is called the *structure tree* of G, defined as follows. (Fig. 1 (d) shows an example of the structure tree[10].)

$V[T] := V_0 \cup V_1 \cup \cdots \cup V_k \cup \{r\}$, where r is the (artificial) root of T. Each $v \in V_0$ is a leaf of T, and a vertex $v \in V_i$ ($i \in \{0, \ldots, k\}$) is on the level i of T, i.e., $v \in V_i$ ($i \geq 1$) is the parent of $u \in V_{i-1}$ if "v is made by contracting a subset (an isolated-clique or a double-isolated-clique) $Q \subseteq V_{i-1}$ such that $u \in Q$" or "$v = u$ (i.e., u is not included in an isolated clique in G_{i-1})." The root r is the parent of every vertex in V_k. (The reason r is added is only to make T a tree.)

We introduce a function $W : V[T] \backslash \{r\} \to 2^V$ and coloring on the vertices in $V[T]$ as follows:

- For $v \in V_0$:
  - $W(v) = \{v\}$, and
  - if $d(v) > \delta$, then v is colored *red*, otherwise uncolored.
- For $v \in V_i$ ($i = 1, \ldots, k$):
  - let $S(v)$ be the set of uncolored children of v,

[10] In this example, we ignore to color vertices red. Since $\delta < 4.5$ follows $\delta/\epsilon = 4.5$, some vertices in this figure might have to be red.

- $W(v) = \bigcup_{u \in S(v)} W(u)$, and
- if $|W(v)| > \delta/\epsilon$, then v is colored *blue*,
- else if $v \in V_k$ and $W(v) \neq \emptyset$, then v is colored *yellow*,
- otherwise, v is uncolored.

Note that for any two distinct colored vertices $u, v \in V[T]$, $W(u) \cap W(v) = \emptyset$. For every $v \in V[T]$, we also define a weight function as $w(v) = |W(v)|$. For a blue (resp. yellow) colored vertex $v \in V[T]$, $W(v) \subseteq V$ is called a *blue (resp. yellow) component*.

By using these colors, we also color the edges in E $(= E_0)$ in the following manner:

- For every red vertex $v \in V_0$ $(= V)$, all edges in $E_G(v)$ are colored *red*.
- For every blue component $W \subseteq V$, for every edge $e \in E_G(W)$, if e is not colored red, then e is colored *blue*.
- For every yellow component $W \subseteq V$, for every edge $e \in E_G(W)$, if e is not colored either red or blue, then e is colored *yellow*.

The other edges in E are uncolored. The set of red, blue, and yellow edges in E are represented by R, B, and Y, respectively. These colors are preserved in $G_1 = \mathcal{E}(G_0)$, $G_2 = \mathcal{E}(G_1)$, …, $G_k = \mathcal{E}(G_{k-1})$, e.g., if an edge $e \in E_i$ is red, then the corresponding edge in E_{i+1} is also red.

2.3 Proof of Theorem 1.9

Before showing the proof of Theorem 1.9, we prepare some lemmas.

▶ **Lemma 2.3.** *For any G_i $(i \in \{0, \dots, k\})$, all edges incident to a vertex with a degree higher than δ are red.*

Proof. For $G_0 = G$, the statement clearly holds from the coloring rule. 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. ◀

▶ **Lemma 2.4.** $|R|, |B| < \epsilon n$*,* $|Y| < \delta n_0/2$*.*

Proof. $|R| < \epsilon n$ is directly obtained from Lemma 2.1. Let $v \in V_i$ be a blue vertex such that a non-red edge exists in $E(W(v))$. From Lemma 2.3, $d(W(v)) \leq \delta$. Thus $d(W(v))/w(v) < \delta/(\delta/\epsilon) = \epsilon$. This means that the average number of blue edges per a vertex is less than ϵ. Therefore $|B| < \epsilon n$. From Lemma 2.3, all edges incident to a vertex with degree higher than δ 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 $|V_k| < n_0$, we have $|Y| < \delta n_0/2$. ◀

Let v_1^R, …, $v_{k_r}^R$ be the red vertices (k_r is the number of red vertices). Let W_1^B, …, $W_{k_b}^B$ be the blue components (k_b is the number of blue components). Let W_1^Y, …, $W_{k_y}^Y$ be the yellow components (k_y is the number of yellow components). We consider a family of vertex subsets as

$$\mathcal{P} := \{\{v_i^R\} \mid i = 1, \dots, k_r\} \cup \{W_i^B \mid i = 1, \dots, k_b\} \cup \{W_i^Y \mid i = 1, \dots, k_y\}.$$

From the definition of the function W and the coloring, $\mathcal{P}$ is clearly a partition of V.

Now we can prove Theorem 1.9.

Proof of Theorem 1.9. If $n \leq \delta n_0/(2\epsilon)$, then the statement is clear by setting $t \geq \delta n_0/(2\epsilon)$. Thus, we assume that $n > \delta n_0/(2\epsilon)$. Let G' be a graph obtained by deleting all red, blue, and yellow edges from G. From Lemma 2.4, the number of deleted edges is less than

$$2\epsilon n + \delta n_0/2 < 3\epsilon n. \tag{2}$$

Next, we will show that the maximum size of connected components in G' is at most $\delta(\delta+1)/\epsilon$. Assume that there exists a connected component $G'(X) = (X, E_X)$ consisting of more than $\delta(\delta+1)/\epsilon$ vertices in G'. X includes no vertex v with $d_G(v) > \delta$, since from Lemma 2.3 all edges in $E_G(v)$ are colored red. Moreover, there is no blue component $W \subseteq V$ such that $X \cap W \neq \emptyset$ and $X \backslash W \neq \emptyset$, as otherwise X would be disconnected in G' (by deleting blue edges).

From this it follows that there is a blue or yellow component $W = W(x)$ such that $X \subseteq W(x)$. If x is a yellow vertex, then $w(x) \leq \delta/\epsilon$ (as otherwise x would be colored blue), and $|X| \leq w(v) \leq \delta/\epsilon < \delta(\delta+1)/\epsilon$, which is a contradiction. Thus x must be a blue vertex. Assume that $x \in V_h$. Let $Z \subseteq V_{h-1}$ be the set of children of x (in T). Z consists of an isolated clique or a double-isolated-clique in G_{h-1}. Let $S(x) (\subseteq Z)$ be the set of uncolored vertices in Z. For every vertex $v \in S(x)$, $d_{G_{h-1}}(v) \leq \delta$ (from Lemma 2.3). From this and the fact that Z consists of an isolated clique or a double-isolated-clique, it follows that $|Z| \leq \delta + 1$.

For $v \in S(x)$, $w(v) \leq \delta/\epsilon$. Hence,

$$w(x) = \sum_{v \in S(x)} w(v) \leq |S(x)| \cdot \delta/\epsilon \leq |Z| \cdot \delta/\epsilon \leq (\delta+1)\delta/\epsilon,$$

which is a contradiction. Therefore, the maximum size of connected components in G' is $\delta(\delta+1)/\epsilon$.

Thus, we have proved that G is $(\max\{\delta n_0/(2\epsilon), \delta(\delta+1)/\epsilon\}, 3\epsilon)$-hyperfinite. Here, ϵ is an arbitrary real number in $(0, 1]$, then by defining $t_{1.9} = \max\{3\delta n_0/(2\epsilon), 3\delta(\delta+1)/\epsilon\}$, G is $(t_{1.9}, \epsilon)$-hyperfinite for any $\epsilon > 0$. ◀

3 Testing Algorithm

3.1 Deterministic partitioning oracle

The global partitioning algorithm of Theorem 1.9 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 [20] 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[11]:

▶ **Definition 3.1.** $\mathcal{O}$ 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

[11] However, since Levi and Ron's algorithm [20] looks fast, using it may be better in practice.

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 3.2.** *There is a deterministic $(t_{1.9}, \epsilon)$-partitioning oracle $\mathcal{O}_{HSF}$ for HSF with $\gamma > 2$ with query complexity $\delta^{O(\delta^2/\epsilon + n_0)}$ for one query.*

Before giving a proof of this lemma, 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 $N(d, t)$.

Proof of Lemma 3.2. The global algorithm of Theorem 1.9 can be easily simulated locally. To find $\mathcal{P}(v)$, if $d(v) > \delta$, then the algorithm outputs $\mathcal{P}(v) := \{v\}$. Otherwise, if the algorithm finds a vertex u with $d(u) > \delta$ in the process of the local search, u is ignored (the algorithm does not check the neighbors of u). Thus, the algorithm behaves as on the bounded-degree model. For any vertex v, $|\mathcal{P}(v)| \leq t_{1.9} = O(\delta^2/\epsilon)$. Each $u \in B_G(v, \delta, t_{1.9})$ may be included in $\mathcal{P}(w)$ of $w \in B_G(u, \delta, t_{1.9})$. Then, the algorithm checks most vertices in $B_G(v, \delta, 2t_{1.9}) = B_G(v, \delta, O(\delta^2/\epsilon + n_0))$, and thus the query complexity for one call of $\mathcal{P}(v)$ is at most $\delta^{O(\delta^2/\epsilon + n_0)}$. ◄

3.2 Abstract of the algorithm

The method of constructing a testing algorithm based on the partitioning oracle of Lemma 3.2 is almost the same as the method used in [23]. We use a distribution vector, which will be defined in Definition 3.3, of rooted subgraphs consisting of at most a constant number of vertices.

► **Definition 3.3.** 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 $||v - u||_1 \leq \epsilon$.

By using the same discussion as in Theorem 3.1 in [23], the following lemma is proven.

► **Lemma 3.4.** *There exist functions $\lambda_{3.4} = \lambda_{3.4}(\mathcal{HFS}, \epsilon)$, $d_{3.4} = d_{3.4}(\mathcal{HFS}, \epsilon)$, $t_{3.4} = t_{3.4}(\mathcal{HFS}, \epsilon)$, $N_{3.4} = N_{3.4}(\mathcal{HFS}, \epsilon)$ such that for every $\epsilon > 0$ the following holds: For every $G_1, G_2 \in \mathcal{HFS}$ on $n \geq N_{3.4}$ vertices, if $|freq_{G_1}(d_{3.4}, t_{3.4}) - freq_{G_2}(d_{3.4}, t_{3.4})| \leq \lambda_{3.4}$, then G_1 and G_2 are ϵ-close.* ◄

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 1.9. This is done locally (shown by Lemma 3.2). 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 [23] 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 $d_{3.4}$, all edges incident to v are ignored. By adding this process, G is regarded as $G|d_{3.4}$. This modification does not effect the result by Lemma 2.1. 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 $d_{3.4}$ is finite (bounded by $O({d_{3.4}}^{n^2})$).

Proof of Theorem 1.10. Obtained from the above discussion. ◂

4 Summary and future work

We presented a natural class of multigraphs $\mathcal{HSF}$ representing scale-free networks, and we showed that a very wide subclass of it is hyperfinite (Theorem 1.9). By using this result, the useful result that every property is testable on the class (Theorem 1.10) is obtained.

$\mathcal{HSF}$ is a class of multigraphs based on the hierarchical structure of isolated cliques. We may relax “isolated cliques” to “c-isolated cliques” or “isolated dense subgraphs [15]” and we may introduce a wider class. We consider such classes also to be hyperfinite. Finding such classes and proving their hyperfiniteness is important future work.

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 Numbers 24650006 and 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., 39(1):143–167, 2009.

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. 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 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).

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

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

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

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

20 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).)

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

22 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.

23 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.)

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

25 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.

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

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