Maximum Leaf Spanning Trees of Growing Sierpinski Networks Models

Bing Yao[1], Xia Liu[1], Jin Xu[2]

1. College of Mathematics and Statistics, Northwest Normal University, Lanzhou, 730070, China

2. School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, China

Abstract

The dynamical phenomena of complex networks are very difficult to predict from local information due to the rich microstructures and corresponding complex dynamics. On the other hands, it is a horrible job to compute some stochastic parameters of a large network having thousand and thousand nodes. We design several recursive algorithms for finding spanning trees having maximal leaves (MLS-trees) in investigation of topological structures of Sierpinski growing network models, and use MLS-trees to determine the kernels, dominating and balanced sets of the models. We propose a new stochastic method for the models, called the edge-cumulative distribution, and show that it obeys a power law distribution. **Keywords:** Spanning trees, scale-free, Sierpinski, algorithm

PACS 89.75.Da, 05.45.Df, 02.10.Ox, 89.75.Fb

1   Introduction

In understanding complex networks, one must know the global properties of networks as well as the local properties such as the degree distribution. Bollobás and Riordan Ref.[2] consider a random graph process in which vertices are added to the graph one at a time and joined to a fixed number m of earlier vertices, where each earlier vertex is chosen with probability proportional to its degree. This process was introduced by Barabási and Albert in Ref. [1], as a simple model of the growth of real-world graphs such as the world-wide web.

It is well known that the dynamical phenomena of networks such as traffic and information flow are very difficult to predict from local information due to the rich microstructures and corresponding complex dynamics (Ref. [5]). On the other hands, it is difficult to implement miscellaneous measurements on complex networks, such as the betweenness centrality (BC) $b(\bullet) = \sum_{i \neq j} \frac{b(i, \bullet, j)}{b(i,j)}$, where $\bullet$ is a node or an edge, and $b(i, \bullet, j)$ is the number of shortest (i, j)-paths through $\bullet$ and $b(i, j)$ is the number of shortest (i, j)-paths. Clearly, this is a horrible job to compute the betweenness centrality of a network having thousand and thousand nodes. But Kim *et al.* (Ref. [5]), after studying the properties of the spanning trees with maximum total edge betweenness centrality, point out that the scale-free spanning trees represent the *communication kernels* on networks, and the scale-free spanning trees show robust characteristics in the degree correlation and the betweenness centrality distribution. They defined the communication kernel of a network as the spanning tree with a set of edges maximizing the summation of their edge BC's on the original networks, and investigated the structural and dynamical properties of the spanning tree of complex networks

and the role of shortcuts in the networks, and find that the spanning trees show scale-free behavior in the degree distributions.

As known, the Maximum Leaf Spanning Tree (MLS-tree) problem, which asks to find, for a given graph, a spanning tree with as many leaves as possible, is one of the classical NP-complete problems in Ref. [8]. Fernau *et al.* (Ref. [7]) investigated MLS-trees based on an exponential time viewpoint that is equivalent to the Connected Dominating Set problem (CDSP), and present a branching algorithm whose running time of $O(1.8966^n)$ has been analyzed using the Measure-and-Conquer technique as well as a lower bound of $\Omega(1.4422^n)$ for the worst case running time of their algorithm. By means of MLS-trees of growing networks in Ref. [11], a *stochastic network model* can be defined as $M_t = (p(u, k, t), G(t), UG)$ for $t \in [a, b]$, where UG is the *underlying graph* of M_t that contains all nodes and links appeared in M_t for $t \in [a, b]$; $p(u, k, t)$ is the *probability* of a node u being connected with other k nodes in $M_{t'}$ $(t' \in [a, t))$; $G(t)$ is the *connected topological graph* of M_t. A node of M_t is called an *alltime-hub* node if it is not a leaf of any MLS-tree of M_t at $t \in [a, b]$. A *kernel* of M_t is an induced graph over the set of alltime-hub nodes.

We show our algorithms to find MLS-trees for investigating topological structures of growing Sierpinski network models (GSN-models) introduced in Ref. [13]. All graphs mentioned here are simple, undirected and finite. A *leaf* is a node of degree one. For a graph G, we let $L(G)$ stand for the set of its leaves, and $D(G)$ be the diameter of G, that is, $D(G) = \min_{i \neq j}\{d(i, j)\}$, where $d(i, j)$ is the *geodesic distance* from node i to node j. Notation $|X|$ is the number of elements of a set X.

2   GSN-models

The graph O in Figure 1 shows the result of a *fractal-operation* that will be used in the following. For a given triangle ΔABC with three nodes A, B, C in the plane, we embed another triangle Δabc with three nodes a, b, c in the inner face of ΔABC, and then join A with b and c to produce two edges Ab, Ac; join B with c and a to form two edges Bc, Ba; and join C with a and b to generate two edges Ca, Cb. The resulting graph O is called the *base*, and it has six inner triangles that bound six regions $I, III' \, II, I', III, II'$ in the clockwise direction. For simpler statement, we call A, B, C three *major nodes* of the base O, where A is called the *left major node*, B the *top major node*, and C the *right major node*, and we call a, b, c to be three *submajor nodes*. We restate the construction of the networks $N(t)$ shown in Ref. [13] by adding a labelling function below.

2.1   Construction of GSN-models

Let $N(0)$ be the initial network pictured in Figure 1, and let $V(0)$ be the node set of $N(0)$. We define a labelling f such that $f(\alpha) = 0$ for each node $\alpha \in V(0)$. Do a fractal-operation to the inner face of $N(0)$ by adding a new triangle produces the second GSN-model $N(1)$, and label $f(\beta) = 1$ for every node $\beta \in V(1) \setminus V(0)$. To form the third GSN-model $N(2)$ from $N(1)$, we do a fractal-operation to each inner triangle Δuvw of $N(1)$ without $f(u) = f(v) = f(w)$, and label each node
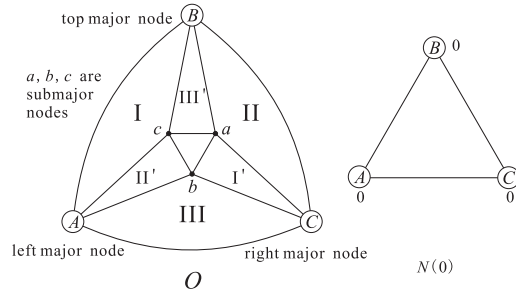
Figure 1: The left is for illustrating a fractal-operation, and the right is the initial network.

$x \in V(2) \setminus V(1)$ as $f(x) = 2$. Go on in this way, every GSN-model $N(t)$ can be obtained from the previous GSN-model $N(t-1)$ for $t \geq 2$ by doing a fractal-operation to each inner triangle Δxyz of $N(t-1)$ without $f(x) = f(y) = f(z)$, and label each node $w \in V(t) \setminus V(t-1)$ as $f(w) = t$.

Let $n_v(t)$ and $n_e(t)$ be the numbers of nodes and edges of the network $N(t)$, respectively. Clearly, each $N(t)$ has an outer face ΔABC and an inner face Δabc for $t \geq 1$ (see Figure 2). Another way to generate $N(t)$ from $N(t-1)$ is the generalized MLS-TREE algorithm introduced later.
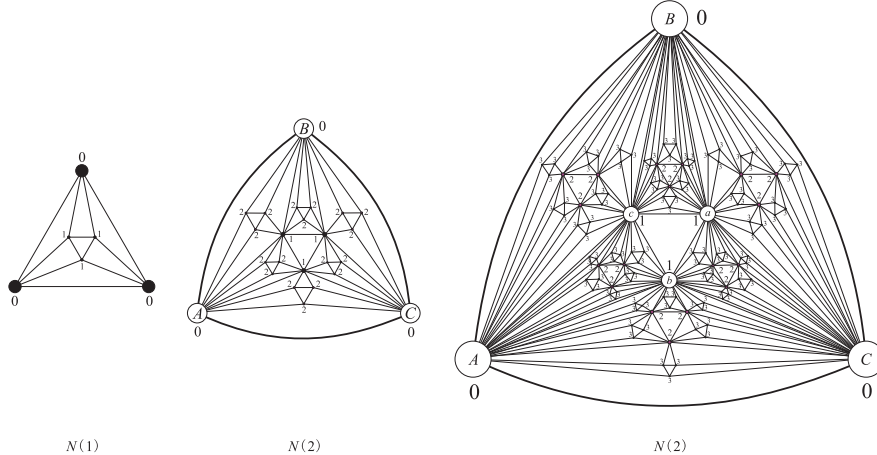

Figure 2: Three GSN-models $N(1)$, $N(2)$ and $N(3)$.

3   Deterministic statistics of GSN-models

Four first GSN-models $N(0)$, $N(1)$, $N(2)$ and $N(3)$ are shown in Figures 1 and 2, respectively. Notation $n_d(t)$ stands for the number of nodes of degree d and let $\Delta(t)$ be the maximum degree in $N(t)$. For $t \geq 2$, the *degree spectrum* of $N(t)$ is that each number $n_d(t)$ of nodes of degree $d = 1 + 3^k$ is equal to $3 \cdot 6^{t-k}$ for $k = 1, 2, \ldots, t-1$, respectively; and the number of nodes of maximum degree $\Delta(t) = 1 + 3^t$ is $n_{\Delta(t)}(t) = 6$ that are the major nodes A, B, C and the subnodes a, b, c. It is easy to show $n_v(t)$ in the formula (1) by the degree spectrum of $N(t)$. Since $N(t)$ is a maximal planar

graph, so $n_e(t) = 3n_v(t) - 6$,

$$n_v(t) = (3 \cdot 6^t + 12)/5, \quad n_e(t) = (9 \cdot 6^t + 6)/5 \tag{1}$$

as well as there are $n_e(t) - n_v(t) + 1$ inner triangles in $N(t)$ by the famous Euler's formula on planar graphs. By the degree spectrum of $N(t)$ and by the *linear preferential attachment rule* (Ref.[1], [4]), we can get the probability of joining a new node u out of $N(t)$ to a node $v_{d(t)}$ of degree $d(t)$ in $N(t)$ as follows, $d(t) = k$,

$$P(u \to v_{d(t)}) = \frac{k(v_{d(t)})}{\sum_{w \in V(t)} k(w)} = \frac{3^k + 1}{2n_e(t)} \approx \frac{5}{18} \cdot \frac{3^k}{6^t}, \tag{2}$$

where $k(x)$ is the degree of node x in $N(t)$. Again we obtain $3^{t-k} \cdot P(u \to v_{d(t)}) = P(u \to v_{\Delta(t)})$ for large integers $t > 0$, which means that the nodes of large degrees play the role like hubs connecting the whole network together. This phenomenon is known as "the rich get richer" paradigm.

The *average (mean) degree* $\langle k \rangle$ of $N(t)$ is defined as

$$\langle k \rangle = \frac{2n_e(t)}{n_v(t)} = \frac{1}{n_v(t)} \sum_{v \in V(t)} k(v). \tag{3}$$

The *average-square (mean-square) degree* $\langle k^2 \rangle$ of $N(t)$ is determined by $\langle k^2 \rangle = \frac{1}{n_v(t)} \sum_{v \in V(t)} k^2(v)$. Newman [10] pointed that a giant component exists in the network if and only if $\langle k^2 \rangle - 2\langle k \rangle > 0$. We verify that $N(t)$ holds $\langle k^2 \rangle - 2\langle k \rangle > 0$ when $t \geq 2$.

3.1 The edge-cumulative distribution

Motivated from the cumulative degree distribution that is an important character of scale-free networks (Refs. [9], [13]), we propose a deterministic statistic for $2 < t_i < t$, named as the *edge-cumulative distribution* $P_{\text{e-cum}}(k)$, as follows

$$P_{\text{e-cum}}(k) = \frac{1}{n_e(t)} \sum_{j=0}^{t_i} n_e(j)$$

$$= \frac{1}{9 \cdot 6^t + 6} \sum_{j=0}^{t_i} (9 \cdot 6^j + 6) \tag{4}$$

$$= \frac{1}{9 \cdot 6^t + 6} \left[15 + 6t_i + \frac{54}{5}(6^{t_i} - 1) \right] \approx \frac{6}{5} 6^{t_i - t}$$

Plugging $t_i = t - \frac{\ln k}{\ln 3}$ into Eq. (4) leads to $P_{\text{e-cum}}(k) \propto \frac{6}{5} k^{-1 - \ln 2/\ln 3}$, which means that $P_{\text{e-cum}}(k)$ follows a power law form with the exponent $\gamma_k = 1 + \frac{\ln 2}{\ln 3}$.

3.2 (α_k, β_k)-GSN models

By the degree spectrum of a GSN-model $N(t)$, for $4 \leq d \leq 3^k + 1$, adding numbers of nodes of degree $d \leq k$ together is $S(\leq k) = \sum_{d \leq k} n_d(t) = \sum_{i=1}^{k} 3 \cdot 6^{t-i} = \frac{3}{5} 6^{t-k}(6^k - 1)$, and adding degrees of

nodes of degree $d \leq k$ is equal to $D(\leq k) = \sum_{d \leq k} d \cdot n_d(t) = \sum_{i=1}^{k} 3 \cdot 6^{t-i}(3^i + 1) = \frac{3}{5}6^t(6 - \frac{1}{6^k} - \frac{5}{2^k})$. Thereby, the sum $S(\geq k+1)$ of numbers of nodes of degree d with $3^{k+1} + 1 \leq d \leq 3^t + 1$ is $S(\geq k+1) = n_v(t) - S(\leq k)$, and the sum $D(\geq k+1)$ of their degrees is equal to $D(\geq k+1) = 2n_e(t) - D(\leq k)$. If the *node-number proportion* $\frac{S(\geq k+1)}{n_v(t)} = \alpha_k$, so we have

$$\alpha_k = \frac{3 \cdot 6^{t-k} + 12}{3 \cdot 6^t + 12} \quad \sim \quad 6^k = \frac{1}{\alpha_k}. \tag{5}$$

we get $k = -\frac{\ln \alpha_k}{\ln 6}$. From the node-number proportion $\frac{S(\leq k)}{n_v(t)} = 1 - \alpha_k$, we solve $k = -\frac{\ln \alpha_k}{\ln 6}$ too. We call $N(t)$ an (α_k, β_k)-*GSN-model*, where $\beta_k = \frac{D(\geq k+1)}{2n_e(t)} = 1 - \frac{D(\leq k)}{2n_e(t)}$, and furthermore $S(\geq k+1)D(\geq k+1) = 2\alpha_k \beta_k n_v(t)n_e(t)$. As a test, we take $\alpha_k = \frac{1}{2} \cdot \frac{1}{10^6}$, so $k \approx 8.0974$ and the *node-degree proportions* are

$$\frac{D(\leq k)}{2n_e(t)} \approx 1 - \frac{1}{6}\left(\frac{1}{6^k} + \frac{5}{2^k}\right) \approx 0.997, \tag{6}$$

and $\beta_k \approx 0.003$. The parameters α_k, β_k show a description of a GSN-model as: The nodes having degrees$\leq k$ show a powerful controlling almost edges of $N(t)$ as k is smaller. Conversely, the nodes with degrees$\geq k+1$ are incident to fewer edges of $N(t)$, however, they connect the the network model together. For example, deleting the nodes of $N(k)$ from $N(t)$ with $t > k \geq 1$ produces 6^{k-1} fragments. Moreover, the deletion of nodes of $V(t-1)$ from $N(t)$ make the remainder consisting of 6^{t-1} components (triangles) in which the total number of nodes is equal to $3 \cdot 6^{t-1} = 5n_v(t-1) - 12$. Obviously, such a deletion destroys the network made by a GSN-model into pieces, since $3 \cdot 6^{t-1}/n_v(t) = \frac{5}{6}$.

4   MLS-trees of GSN-models

In this section we will determine the *kernels* of GSN-models by MLS-trees in the models, and furthermore we present several algorithms for finding MLS-trees. The notation $T^M(t)$ denotes an MLS-tree of $N(t)$, and $L(T^M(t))$ stands for the set of leaves of $T^M(t)$. It is easy to verify that $|L(T^M(0))| = 2$, $|L(T^M(1))| = 4$ and $|L(T^M(2))| = 19$.

4.1   Construction of MLS-trees of GSN-models

Let $M_{st}(t)$ be the set of MLS-trees of $N(t)$ such that three major nodes A, B, C of every MLS-tree of $M_{st}(t)$ are not leaves, and every one of $M_{st}(t)$ has two edges AB, BC and has no the edge AC. Since an MLS-tree of $N(t)$ can be constructed by different MLS-trees of $N(t-1)$, so we call them *non-uniformly MLS-trees*.

Generalized MLS-TREE algorithm

Input: A GSN-model $N(t)$ for $t \geq 3$, $M_{st}(2) = \{T_i^M(2) : i = 1, 2, \ldots, m_2\}$, where each i is called a *footscript*, and m_2 is the number of elements of $M_{st}(2)$.

Output: $M_{st}(t)$.

1. $M_{st}(2) \leftarrow M_{st}(2)$, $P_2 \leftarrow \{(k_1, k_2, \ldots, k_6)\}$, where $(k_1, k_2, \ldots, k_6)$ are permutations over all footscripts of MLS-trees of $M_{st}(2)$; $i \leftarrow 2$.

2. If $i < t$, $M_{st}(i + 1) \leftarrow \emptyset$, go to 3; otherwise $M_{st}(t) \leftarrow M_{st}(i)$, go to 5.

3. If $P_i \neq \emptyset$, go to 4; otherwise $P_{i+1} \leftarrow \{(k_1, k_2, \ldots, k_6)\}$, where $(k_1, k_2, \ldots, k_6)$ is a permutation of footscripts of MLS-trees of $M_{st}(i)$; $i \leftarrow i + 1$ go to 4.

4. Take a permutation $Q \in P_i$, **do** (Ref. the graph O of Figure 1 and six graphs shown in Figure 3):

(a1) rename $T_{k_1}^M(i) \in M_{st}(i)$ as I-$T_{k_1}^M(i)$ and set its left major node $c \leftarrow A$, its top major node $A \leftarrow B$, and its right major node $B \leftarrow C$;

(a2) delete two edges AB, BC of $T_{k_2}^M(i) \in M_{st}(i)$, and name the remainder as III'-$T_{k_2}^M(i)$, and then set its left major node $c \leftarrow A$, its top major node $B \leftarrow B$, and its right major node $a \leftarrow C$;

(a3) rename $T_{k_3}^M(i) \in M_{st}(i)$ as II-$T_{k_3}^M(i)$, and set its left major node $a \leftarrow A$, its top major node $B \leftarrow B$, and its right major node $C \leftarrow C$;

(a4) delete two edges AB, BC of $T_{k_4}^M(i) \in M_{st}(i)$ and, name the remainder as I'-$T_{k_4}^M(i)$, and set its left major node $a \leftarrow A$, its top major node $C \leftarrow B$, and its right major node $b \leftarrow C$;

(a5) rename $T_{k_5}^M(i) \in M_{st}(i)$ as III-$T_{k_5}^M(i)$, and set its left major node $b \leftarrow A$, its top major node $C \leftarrow B$, and its right major node $A \leftarrow C$;

(a6) delete its two edges AB, BC of $T_{k_6}^M(i) \in M_{st}(i)$ and, name the remainder as II'-$T_{k_6}^M(i)$, and set its left major node $b \leftarrow A$, its top major node $A \leftarrow B$, and its right major node $c \leftarrow C$.

Identify the major nodes of the above six graphs I-$T_{k_1}^M(i)$, III'-$T_{k_2}^M(i)$, II-$T_{k_3}^M(i)$, I'-$T_{k_4}^M(i)$, III-$T_{k_5}^M(i)$ and II'-$T_{k_6}^M(i)$ having the same letters into one node, respectively. The resulting is just an MLS-tree $T^M(i+1)$ of $N(i+1)$, and $T^M(i+1)$ has three submajor nodes a, b, c and three major nodes A, B, C (Ref. Figure 4).

Let $f(A) \leftarrow 0$, $f(B) \leftarrow 0$, $f(C) \leftarrow 0$, $f(a) \leftarrow 1$, $f(b) \leftarrow 1$, $f(c) \leftarrow 1$; $f(x) \leftarrow f(x) + 1$ for $x \in V(T^M(i+1)) \setminus \{a, b, c, A, B, C\}$; $M_{st}(i+1) \leftarrow M_{st}(i+1) \cup \{T^M(i+1)\}$, $P_i \leftarrow P_i \setminus \{Q\}$, go to 3.

5. return $M_{st}(t)$.

Theorem 1. *Every MLS-tree of a GSN-model $N(t)$ for $t \geq 3$ has $19 \cdot 6^{t-2}$ leaves.*

Proof. It is not hard to verify that each MLS-tree of $N(t)$ has $19 \cdot 6^{k-2}$ leaves for $k = 2, 3$. So, we can use the induction on time steps $t \geq 3$. In the generalized MLS-TREE algorithm, we can select an MLS-tree $T^M(3)$ having $19 \cdot 6^{3-2}$ leaves and take $k_i = k_j$ for all permutations $(k_1, k_2, \ldots, k_6)$ at each time step to obtain an MLS-tree $T^M(t)$ having $19 \cdot 6^{t-2}$ leaves for $t \geq 4$. Note that $N(t)$ is the resulting of overlapping six models $N_i(t-1)$ ($\cong N(t-1)$) for $i = 1, 2, \ldots, 6$, according to the generalized MLS-TREE algorithm. If $N(t)$ has a spanning tree G such that $|L(G)| > 19 \cdot 6^{t-2}$,
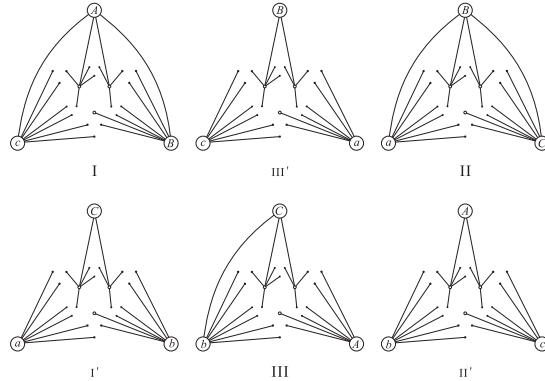
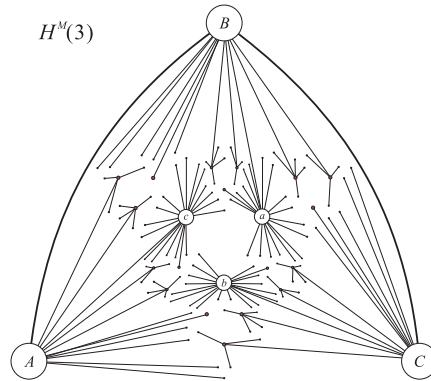Figure 3: Six graphs for illustrating the generalized MLS-TREE algorithm.

Figure 4: An MLS-tree $H^M(3)$ obtained by six graphs shown in Figure 3 and the generalized MLS-TREE algorithm.

so G induces six spanning trees G_i of $N_i(t-1)$ for $i = 1, 2, \ldots, 6$ by the generalized MLS-TREE algorithm. We have one spanning tree G_i with $|L(G_i)| > 19 \cdot 6^{t-2}$, which contradicts with the induction hypothesis. $\square$

The number $|L(T^M(t))|$ of leaves of a MLS-tree $T^M(t)$ of $N(t)$ is $19 \cdot 6^{t-2} = \frac{95}{18}n_v(t-1) - \frac{38}{3}$, and also greater than the number $n_v(t) - n_v(t-1)$ of adding new nodes to $N(t-1)$ since $19 \cdot 6^{t-2} = 6^{t-2} + n_v(t) - n_v(t-1)$.

4.2   Balanced sets in GSN-models

In Ref. [12], a non-empty node subset X of node set $V(t)$ of a GSN-model $N(t)$ is called a T^M-*balanced set* if

$$|X \cap T_i^M(t)| = |X \cap T_j^M(t)| \tag{7}$$

holds for any pair of MLS-trees $T_i^M(t)$ and $T_j^M(t)$ of $N(t)$. Obviously, $V(t)$ holds Eq.(7) true, but it is trivial. Our goal is to find T^M-balanced sets $X \neq V(t)$, and such sets are called *proper*

T^M-balanced sets.

Theorem 2. *For every GSN-model $N(t)$ with $t \geq 3+k$ and $k \geq 0$, each node of $N(k)$ is not a leaf of any MLS-tree of $N(t)$.*

Proof. It is not hard to see that $V(0)$ is not a proper T^M-balanced set of $N(1)$ and $N(2)$ (Ref. these two models $N(1)$ and $N(2)$ shown in Figure 2).

Case 1. $k = 0$ and $t \geq 3$. For any MLS-tree $T^M(t)$ of $N(t)$, we say $V(0) \cap L(T^M(t)) = \emptyset$ when $t \geq 3$. If it is not so, without loss of generality, the left major node $A \in V(0) \cap L(T^M(t))$, namely, the node A is a leaf of a MLS-tree $T^M(t)$ of the GSN-model $N(t)$. In the region II', there are triangles $\Delta x_i^{(r)} y_i^{(r)} z_i^{(r)}$ of $N(r)$ holding distances $d(A, x_i^{(r)}) = 2$ and $d(A, y_i^{(r)}) = d(A, z_i^{(r)}) = 1$ for $i = 1, 2, \ldots, 3^{r-1}$ with $2 \leq r \leq t$. In $N(t)$ and $N(t-1)$, there are two triangles $\Delta x_i^{(t)} y_i^{(t)} z_i^{(t)}$ and $\Delta x_j^{(t-1)} y_j^{(t-1)} z_j^{(t-1)}$ such that three nodes of $\Delta x_i^{(t)} y_i^{(t)} z_i^{(t)}$ are only connected with nodes $y_j^{(t-1)}$, $z_j^{(t-1)}$ and A. In other words, nodes $y_j^{(t-1)}$, $z_j^{(t-1)}$ are not leaves of $T^M(t)$. We can make another spanning tree H of $N(t)$ by joining A with $y_i^{(t)}, z_i^{(t)}$, joining $y_j^{(t-1)}$ with $x_i^{(t)}$, and make $z_j^{(t-1)}$ to be a leaf of the new spanning tree. Note that we can do the above work in the rest regions I and III. Eventually, we obtain a spanning tree H^* of $N(t)$ such that $|L(H^*)| \geq 2 + L(T^M(t))$, which violates the definition of $T^M(t)$ having maximal leaves in $N(t)$.

Case 2. $k \geq 1$. The constraint $t \geq 3 + k$ is necessary, because there exists an MLS-tree $T^M(3)$ of the GSN-model $N(3)$ such that $(V(1) \setminus V(0)) \cap L(T^M(3)) \neq \emptyset$. Furthermore, based on $T_1^M(3)$, we can use the generalized MLS-TREE algorithm to obtain an MLS-trees $H(m)$ such that $(V(m-2) \setminus V(m-3)) \cap L(H(m)) \neq \emptyset$ with $m \geq 3$.

By contradiction to show that each node of $V(k)$ is not a leaf of any MLS-tree $T^M(t)$ with $t \geq 3+k$. Assume that $N(t)$ has an MLS-tree $T^M(t)$ such that $V(k) \cap L(T^M(t)) \neq \emptyset$ for some k with $1 \leq k \leq t-3$. So, we have a leaf $x_s^{(k)} \in V(k) \cap L(T^M(t))$ and a triangle $\Delta x_s^{(k)} y_s^{(k)} z_s^{(k)}$. By the structure of $N(t)$, there is a triangle $\Delta x_s^{(k+1)} y_s^{(k+1)} z_s^{(k+1)}$ inside of the triangle $\Delta x_s^{(k)} y_s^{(k)} z_s^{(k)}$ such that these two triangles form a subgraph O^* like the graph O shown in Figure 1. In this case, the position of the node $x_i^{(k)}$ is as the same as the node A in Case 1. In the region II' of O^*, there are triangles $\Delta x_i^{(r)} y_i^{(r)} z_i^{(r)}$ of $N(r)$ holding distances $d(x_s^{(k)}, x_i^{(r)}) = 2$ and $d(x_s^{(k)}, y_i^{(r)}) = d(x_s^{(k)}, z_i^{(r)}) = 1$ for $i = 1, 2, \ldots, 3^{r-k-1}$ with $k+2 \leq r \leq t$. Based on the same proof shown in Case 1, we can get a contradiction with the definition of $T^M(t)$. $\square$

A subset S of node set $V(t)$ of a GSN-model $N(t)$ is called a *dominating set* if every node of $V(t)$ is adjacent to a node of S or belongs to S. By Theorem 2 we can confirm the following results

Theorem 3. *Every GSN-model $N(t)$ with $t \geq k + 3$ and $k \geq 0$ holds:*

(i) *every $V(k)$ is a proper connected T^M-balanced set and induces a connected kernel of $N(t)$;*

(ii) *$V(k) \subset V(T_i^M(t)) \cap V(T_j^M(t))$ for any two MLS-trees $T_i^M(t)$ and $T_j^M(t)$ of $N(t)$; and*

(iii) *$V(t) \setminus L(T^M(t))$ is a connected dominating set of $N(t)$.*

Although the nodes with degrees$\geq k+1$ do not control more edges by Eq. (6), but they are a controlling center in $N(t)$ with $t \geq 3+k$. By Theorem 3 we can find a maximal proper connected T^M-balanced set X_1 of $N(t)$, and then get a connected model I_1 induced over X_1. Next, I_1 has a maximal proper connected T^M-balanced set X_2 that induces a connected model I_2. In this way, we obtain a sequence of models $I_1, I_2, \ldots I_m$ such that $I_{i+1} \subset I_i$, and X_{i+1} is a maximal kernel of I_i for $i = 1, 2, \ldots, m-1$. Clearly, $I_i \subset N(t - 2i)$.

4.3   A dynamic algorithm for finding MLS-trees of GSN-models

There are some algorithms to find spanning trees of networks in Ref. [11]. We will apply the Bread-first Search algorithm (BFSA) introduced in Ref. [3] to make our Dynamic First-first BFSA algorithm (DFF-BFSA algorithm) by the motivation of the linear preferential attachment rule (Ref.[4]). Predecessors' children are searched before successors' children, according to a rule of "priority has priority".

DFF-BFSA algorithm

Input: A GSN-model $N(t)$ for $t \geq 0$.

Output: A spanning tree $T(t)$.

1. For the GSN-model $N(0)$, BFSA outputs a spanning tree $T(0)$ with $V(0) = \bigcup_{j=0}^{m(0)} V_j^{(0)}$ and a level function l such that $l(x) = j$ for $x \in V_j^{(0)}$, and the nodes of $V_j^{(0)}$ are ordered well by BFSA for $j = 1, 2, \ldots, m(0)$.

2. Let nei(x, k) be the neighborhood of a node x of $N(k)$ at time step k. At time step $k+1$, $V(k) = \bigcup_{l=0}^{k} \bigcup_{j=m(l-1)+1}^{m(l)} V_j^{(l)}$ (here, $m(-1) = -1$). Implementing BFSA **do**: For every ordered set $V_j^{(l)} = \{x_{j,1}^{(l)}, x_{j,2}^{(l)}, \ldots, x_{j,m(l,j)}^{(l)}\}$ with $l \leq k$, from $i = 1$ to $i = m(l, j)$, scan $y \in$ nei$(x_{j,i}^{(l)}, k+1) \setminus \{w \in V(t) : l(w)$ exists$\}$, $b \leftarrow l(x_{j,i}^{(l)}) + 1$, $l(y) \leftarrow m(k) + b$, and add y to the ordered set $V_{m(k)+b}^{(k+1)}$ as the last node, add node y and edge $yx_{j,i}^{(l)}$ to the spanning tree $T(k)$ in order to form new spanning tree $T(k+1)$; $m(k+1) \leftarrow \max\{m(k) + l(x) + 1 : x \in V(k)\}$, $V(k+1) \setminus V(k) \leftarrow \bigcup_{j=m(k)+1}^{m(k+1)} V_j^{(k+1)}$; $V(k+1) \leftarrow \bigcup_{l=0}^{k+1} \bigcup_{j=m(l-1)+1}^{m(l)} V_j^{(l)}$, go to 3.

3. If $k + 1 = t$, $T(t) \leftarrow T(k+1)$, go to 4; otherwise go to 2.

4. return $T(t)$ with a level function l.

Theorem 4. *The DFF-BFSA algorithm can find an MLS-tree T of a GSN-model $N(t)$ such that T has $19 \cdot 6^{t-2}$ leaves, diameter $D(T) = 2t$ and maximum degree $\Delta(T) = \Delta(t)$ with $t \geq 2$.*

Proof. Suppose that there is a cycle $C = x_1 x_2 \cdots x_m x_1$ in the graph T obtained by the DFF-BFSA algorithm. Without loss of generality, the level values $l(x_1) < l(x_i)$ for $x_i \neq x_1$ in C. So there are three nodes u, v, w such that $l(u) < l(w)$ and $l(v) < l(w)$, which mean that $w \in$ nei$(u) \setminus$ nei(s) and $w \in$ nei$(v) \setminus N(k)$, but it is impossible since (nei$(u) \setminus$ nei$(s)) \cap (N(v) \setminus N(k)) = \emptyset$. Thereby, T contains no cycle (Ref. $y \in$ nei$(x_{j,i}^{(l)}, k+1) \setminus \{w \in V(t) : l(w)$ exists$\}$). Notice that $N(t)$ is
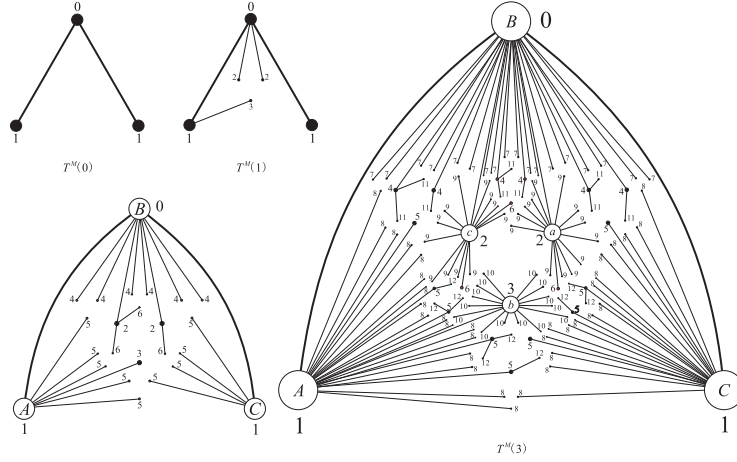
Figure 5: The DFF-BFSA algorithm produces four MLS-trees with level functions in $N(k)$ for $k = 0, 1, 2, 3$.

connected, and at each time step the DFF-BFSA algorithm scans all neighbors of a node. So, T is connected and a spanning tree.

We select the first node $u_0 = B$ in $N(2)$, so $l(u_0 = B) = 0$ and degree $k(u_0)$ is equal to maximum degree $\Delta(k) = 3^k + 1$ of $N(k)$ for $2 \le k \le t$ by the DFF-BFSA algorithm. Notice that $l(A) = l(C) = 1$, so $|k(A) - k(C)| = 1$ and $\max\{k(A), k(C)\} = k(u_0) - 1 = 3^t$. Three nodes A, B, C of the spanning tree T control other $3^{k+1} + 1$ nodes of $N(k)$. For $t = 0, 1, 2$, it is not hard to see diameters $D(T) = 2t$. For $t \ge 3$, by the DFF-BFSA algorithm, every path $P(A, w)$ from node A to a leaf w has at most length $(t - 1)$ if it does not pass through node B, and each path $P(C, w')$ from node C to a leaf w' has at most length $(t - 1)$ if it does not pass through node B. Thereby, the path from w to w' has length $2t$, which means $D(T) = 2t$.

Notice that the spanning trees have $19 \cdot 6^{t-2}$ leaves for $t = 0, 1, 2, 3$ (Ref. Figure 5). We can confirm that for $t \ge 4$, every spanning tree of $N(t)$ obtained by the DFF-BFSA algorithm has $19 \cdot 6^{t-2}$ leaves according to Theorem 2. The theorem is covered. $\square$

5 Conclusion

For determining the kernels of GSN-models we focus on MLS-trees of GSN-models, and show the structures of some MLS-trees by our algorithms. Clearly, all MLS-trees $T^M(i + 1)$ having the shortest diameter $2(i + 1)$ can be constructed by our generalized MLS-TREE algorithm over all MLS-trees having shortest diameter $D(T^M(i)) = 2i$ in $M_{st}(i)$. Although our DFF-BFSA algorithm can not find spanning trees having maximal leaves in any growing network model, however, we verify it for some growing network models, and find out some interesting spanning trees.

Suppose that a triangle Δxbc and another triangle Δbcy have a common edge bc in a maximal planar graph G whose faces are triangular. We remove the edge bc and then join x with y by an

edge, the resulting is still a maximal planar graph, written as G' and say 'flipping the edge bc'. We call the procedure of obtaining G' from G a *flip operation*. Note that every GSN-model $N(t)$ is a maximal planar graph, and "Any pair of maximal planar graphs on n vertices can be transformed into each other by at most $5.2n - 24.4$ flip operations (Ref. *[6]*)." We propose a problem: *For what value of a positive integer m, does rewiring m edges of a GSN-model $N(t)$ by the flip operation produce a scale-free network model?* For larger integers $t > 0$, we guess that a maximal planar graph H^* obtained from a GSN-model $N(t)$ by flipping some edges having ends in $V(t) \setminus V(t-3)$ is scale-free. The above problem leads to a problem of graph theory: *Determine finite maximal planar graphs $G_1, G_2, \ldots$ such that G_{t-1} is a proper subgraph of G_t and each G_t obeys a power law distribution.*

References

[1] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, **286** (1999), 509-512.

[2] Béla Bollobás, Oliver Riordan, The Diameter Of A Scale-Free Random Graph, Combinatorica **24** (1) (2004) 5-34

[3] J.A. Bondy and U.S.R. Murty. Graph Theory. *Springer*, ISBN: 978-1-84628-969-9, e-ISBN: 978-1-84628-970-5, DOI: 10.1007/978-1-84628-970-5, 2008.

[4] L. da F. Costa F. A. Rodrigues G. Travieso P. R. Villas Boas, Characterization of Complex Networks: A Survey of measurements, arXiv:cond-mat/0505185v5 [cond-mat.dis-nn] 16 Aug 2006.

[5] Dong-Hee Kim, Jae Dong Noh, and Hawoong Jeong, Scale-free trees: The skeletons of complex networks, Physical Review E **70**, 046126 (2004).

[6] Mori, R., Nakamoto, A., and Ota, K. Diagonal flips in Hamiltonian triangulations on the sphere. Graphs Combin. **19**, 3 (2003), 413-418.

[7] Henning Fernau, Joachim Kneis, Dieter Kratsch, Alexander Langer, Mathieu Liedloff, Daniel Raible, Peter Rossmanith, An exact algorithm for the Maximum Leaf Spanning Tree problem, Theoretical Computer Science **412** (2011) 6290-6302.

[8] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, Freeman, 1979.

[9] M. E. J. Newman, The structure and function of complex networks, SIAM Review **45**, 167 (2003),167-256.

[10] M. E. J. Newman, Mathematics of networks, in The New Palgrave Encyclopedia of Economics, 2nd edition, L. E. Blume and S. N. Durlauf (eds.), Palgrave Macmillan, Basingstoke (2008).

[11] Bing Yao, Xia Liu, Wan-jia Zhang, Xiang-en Chen, Xiao-min Zhang, Ming Yao, Zheng-xue Zhao. Applying Graph Theory To The Internet of Things. 2013 IEEE International Conference on High Performance Computing and Communications and 2013 IEEE International Conference on Embedded and Ubiquitous Computing, 2354-2361. DOI: 10.1109/HPCC.and.EUC.2013.339

[12] Bing Yao, Zhong-fu Zhang, and Jian-fang Wang, Some results on spanning trees, Acta Mathematicae Applicatae Sinica, English Series **26** (4) (2010) 607-616.

[13] Zhongzhi Zhang, Shuigeng Zhou, Lujun Fang, Jihong Guan and Yichao Zhang. Maximal planar scale-free Sierpinski networks with small-world effect and power-law strength-degree correlation. EPL (Europhysics Letters), 2007, **79**: 38007.