An Innovative Combinatorial Approach for the Spanning Tree Entropy in Flower Network

Raihana Mokhlissi[1](✉), Dounia Lotfi[1], Joyati Debnath[2], and Mohamed El Marraki[1]

[1] LRIT Associated Unit with CNRST (URAC No 29), Faculty of Sciences,
Mohammed V University in Rabat, 1014 Rabat, Morocco
mokhlissiraihana@gmail.com, {lotfi,marraki}@fsr.ac.ma
[2] Winona State University, Winona, MN 55987, USA
jdebnath@winona.edu

Abstract. The spanning tree entropy of a complex network provides a useful insight about its robustness. The most robust network is the network that has the highest entropy. In this paper, we represent construction of a complex network called Flower Network by using two combinatorial approaches: (1) Bipartition and (2) Reduction. We based both methods on geometrical transformation. We also develop topological properties of the network, obtain analytical expression for its number of spanning trees. In the end, we calculate and compare its spanning tree entropy with those for other networks having the same average degree of nodes for estimating a robust network.

Keywords: Entropy · Complex network · Flower network · Bipartition · Reduction · Number of spanning trees

1 Introduction

In nature, most of the complex systems in real life are represented by networks, where the nodes denote the basic constituents of the system and links describe their interaction. The Internet, electric, imaging, telephone calls, social networks and many other systems are now represented by complex networks [1]. There are many different types of these networks and this classification depends on the properties such as nodes degrees, clustering coefficients, shortest paths. Another concern in studying complex network is how to evaluate the robustness of a network and its ability to adapt to changes [2]. In general, the robustness of a network is correlated to its ability to deal with internal feedbacks within the network and to avoid malfunctioning when a fraction of its constituents is damaged.

In this work, we suggest a structural characterization of robustness in terms of network entropy [3], a structural property of the network. This concept is widely used in thermodynamics to measure the systems' efficiency. For complex networks, we use **the entropy of spanning trees** or what is called the

A. El Abbadi and B. Garbinato (Eds.): NETYS 2017, LNCS 10299, pp. 3–14, 2017.
DOI: 10.1007/978-3-319-59647-1_1

asymptotic complexity [4] in order to quantify the robustness and to characterize the structure. It is related to the capacity of the network to withstand random changes in its structure. Graph theory has provided powerful combinatorial tool to calculate this entropy. This tool represented by **spanning tree** [5] helps us to focus on the relationship between the structure and the function of networks. To obtain the entropy of a complex network, first of all, we have to determine its number of spanning trees (The complexity). This number gives us an idea about how many possible topologies that a network can have. It is obtained by computing the determinant of a submatrix of the Laplacian matrix corresponding to the network (Kirchhoff's matrix-tree theorem) [6,7]. However, for a large network, the evaluation of this determinant is very difficult and even impossible. Most of the recent works have tried to find some alternative methods in order to avoid the arduous calculations of the largest determinant as needed by the algebraic method and enumerate the spanning trees for large and complex networks.

In this paper, we perform a process on a specific self-similar complex network, which is called the Flower network [8]. It consists of self-repeating patterns on all length scales. This network can be constructed by using two combinatorial approaches, which highlight the self-similarity of a complex network at different iterations. These techniques are based on geometrical transformations of the original network, by multiplying the number of nodes in the case of the **Bipartition approach** [9], or by multiplying the number of links in the case of the **Reduction approach** [9]. First, we define these combinatorial methods, we cite their properties, we study their complexity and we generalize these results to the case of k-partite and k-reduced networks. Then, we present two models of the Flower network, the first is characterized by two dimensions of self-similarity denoted by **2-Flower network** and the second is the general case denoted by **k-Flower network**. We examine their construction, we analyze their topological properties and we evaluate their complexity by combining Our approaches. Finally, we calculate their spanning trees entropy and we compare it with those for other networks having the same average degree of nodes in order to determine the most robust network between them.

2 Related Work

In this section, we present our approaches and we quote all the theorems, the definitions and the properties which we need to construct the network and calculate its complexity. We use the standard terminologies indistinctly as: "network" and "graph", "node" and "vertex" and "link" and "edge". Let $G = (V_G, E_G, F_G)$ be a simple planar connected graph with V_G is the set of vertices, E_G is the set of edges and F_G is the set of faces. Our approaches are presented as follows:

2.1 The Bipartition Approach

The bipartite graphs belong to one of the simplest families in the graph theory [9] and their representation helps to study and use in various areas such as semantic web, data mining and segmentation of images, etc.

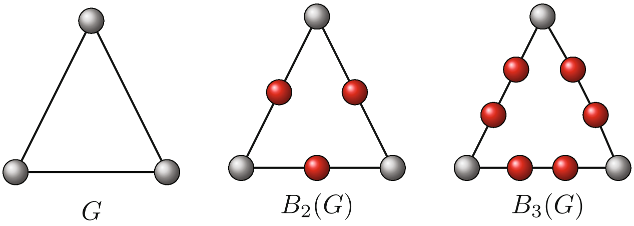

Fig. 1. A graph G, its bipartite graph and its 3-partite graph

Definition 1. *A graph becomes bipartite when we add a new vertex between two directly connected vertices, denoted by $B_2(G)$ (See Fig. 1).*

Lemma 1. *Let G be a planar graph and $B_2(G)$ its bipartite graph. The number of vertices in $B_2(G)$ is given by $V_{B_2(G)} = V_G + E_G$, its number of edges is given by $E_{B_2(G)} = 2E_G$, its number of faces is $F_{B_2(G)} = F_G$ and its average degree is $< z >_{B_2(G)}= \frac{2E_{B_2(G)}}{V_{B_2(G)}} = \frac{4E_G}{V_G+E_G}$.*

Theorem 1. *Let $B_2(G)$ be a bipartite graph of a planar graph G. The number of spanning trees in $B_2(G)$ is given by:*

$$\tau(B_2(G)) = 2^{F_G-1}\tau(G) \tag{1}$$

The main objective of the bipartite approach is to reduce the number of nodes of a network before finding its complexity.

Definition 2. *A k-partite graph of a planar graph G is defined by adding $k-1$ new vertices in each edge, denoted by $B_k(G)$ (See Fig. 1).*

Lemma 2. *Let G be a planar graph and $B_k(G)$ its k-partite graph. The number of vertices in $B_k(G)$ is given by $V_{B_k(G)} = V_G + (k-1)E_G$, its number of edges is given by $E_{B_k(G)} = k \times E_G$, its number of faces is $F_{B_k(G)} = F_G$ and its average degree is $< z >_{B_k(G)}= \frac{2E_{B_k(G)}}{V_{B_k(G)}} = \frac{2k\times E_G}{V_G+(k-1)E_G}$.*

Theorem 2. *Let $B_k(G)$ be a k-partite graph of a planar graph G. The number of spanning trees in $B_k(G)$ is given by:*

$$\tau(B_k(G)) = k^{F_G-1}\tau(G) \tag{2}$$

2.2 The Reduction Approach

The reduction approach is another concept that is characterized by the presence of multiple edges [9]. It is an implementation method for functional programming languages.

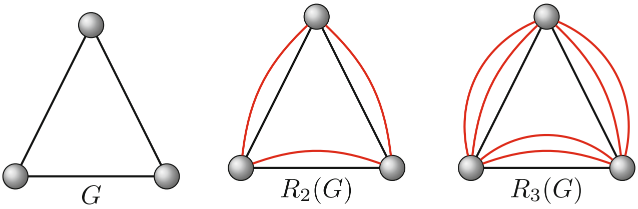

Fig. 2. A graph G, its reduced graph and its 3-reduced graph

Definition 3. *A graph becomes reduced when we add a new edge connecting two existing vertices of a planar graph G. It is denoted by $R_2(G)$ (See Fig. 2).*

Lemma 3. *Let G be a planar graph and $R_2(G)$ its reduced graph. The number of vertices in $R_2(G)$ is given by $V_{R_2(G)} = V_G$, its number of edges is given by $E_{R_2(G)} = 2E_G$, its number of faces is $F_{R_2(G)} = F_G + E_G$ and its average degree is $< z >_{R_2(G)} = \frac{2E_{R_2(G)}}{V_{R_2(G)}} = \frac{4E_G}{V_G}$.*

Theorem 3. *Let $R_2(G)$ be a reduced graph of a planar graph G. The number of spanning trees in $R_2(G)$ is given by:*

$$\tau(R_2(G)) = 2^{V_G - 1}\tau(G) \tag{3}$$

The main objective of the reduction approach is to reduce the number of links of a network and that makes the complexity easy for computation.

Definition 4. *Let G be a planar graph. The k-reduced graph of G noted $R_k(G)$ is obtained when for each pair of vertices of G, we have k multiple edges connecting them (See Fig. 2).*

Lemma 4. *Let G be a planar graph and $R_k(G)$ its k-reduced graph. The number of vertices in $R_k(G)$ is given by $V_{R_k(G)} = V_G$, its number of edges is given by $E_{R_k(G)} = k \times E_G$, its number of faces is $F_{R_k(G)} = F_G + (k-1)E_G$ and its average degree is $< z >_{R_k(G)} = \frac{2E_{R_k(G)}}{V_{R_k(G)}} = \frac{2k \times E_G}{V_G}$.*

Theorem 4. *Let $R_k(G)$ be a k-reduced graph of a planar graph G. The number of spanning trees in $R_k(G)$ is given by:*

$$\tau(R_k(G)) = k^{V_G - 1}\tau(G) \tag{4}$$

3 The Construction and the Topological Properties of the Flower Network

In this section, we introduce the construction of the Flower network which are built in an iterative way. We treat two types of flower graphs: **The 2-Flower graph** based on the reduced and the bipartite approaches (see Fig. 3) and **the k-Flower graph** based on the k-reduced and the k-partite approaches (see Fig. 4). Then, we discuss their topological properties.

3.1 The 2-Flower Graph

The 2-Flower graph denoted by C_n are constructed as follows: For the iteration n = 0, we have a simple edge that connects two vertices. For $n \geq 1$, first, we apply the reduced approach by adding a new multiple link for each edge of the graph in the previous iteration. Then, we apply the bipartite approach to this last obtained graph by adding a new vertex in the middle of each edge. We can denote this process by $C_n = h_2^n(C_0) = h_2(C_{n-1}) = B_2 \circ R_2(C_{n-1})$. For illustration, in Fig. 3, we present 3 iterations of the 2-Flower graph.

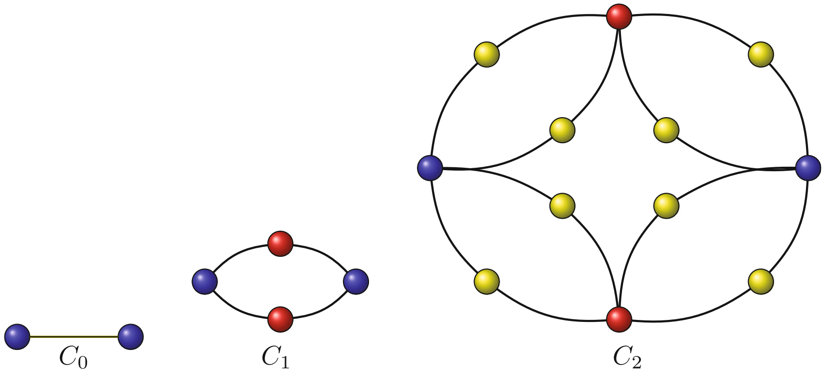

Fig. 3. The 3 iterations of the 2-Flower graph.

According to this construction, the number of edges of C_n is calculated as follows:

$$E_{C_n} = E_{h_2^n(C_0)} = E_{h_2(C_{n-1})} = E_{B_2 \circ R_2(C_{n-1})}$$

Using Lemma 1, we obtain: $E_{C_n} = 2E_{R_2(C_{n-1})}$

And using Lemma 2, we obtain:

$$E_{C_n} = 4E_{C_{n-1}} = 4^2 E_{C_{n-2}} = 4^3 E_{C_{n-3}} = ... = 4^n E_{C_0}$$

So the number of edges of C_n is: $E_{C_n} = 2^{2n}$

Similarly, we find:

The number of vertices of C_n: $V_{C_n} = 2 + \frac{2}{3}(2^{2n} - 1)$

The number of faces of C_n: $F_{C_n} = 1 + \frac{2^{2n}-1}{3}$

The average degree of C_n: $< z >_{C_n} = \frac{2E_{C_n}}{V_{C_n}} = \frac{2^{2n+1}}{\frac{4}{3} + \frac{2^{2n+1}}{3}}$

3.2 The k-Flower Graph:

The k-Flower graph denoted by G$_n$ with n is a number of iterations can be created using the following iterative way: For n = 0, we have a simple edge that connects two vertices. For $n \geq 1$, first, we apply the k-reduced approach to obtain k multiple edges connecting each pair of vertices of the graph in the previous iteration. Then, we apply the k-partite approach to this last obtained graph by adding k−1 vertices in each edge. This process can be denoted by $G_n = h_k^n(G_0) = h_k(G_{n-1}) = B_k \circ R_k(G_{n-1})$. In Fig. 4, we illustrate 3 iterations of the k-flower graph with $k = 3$.

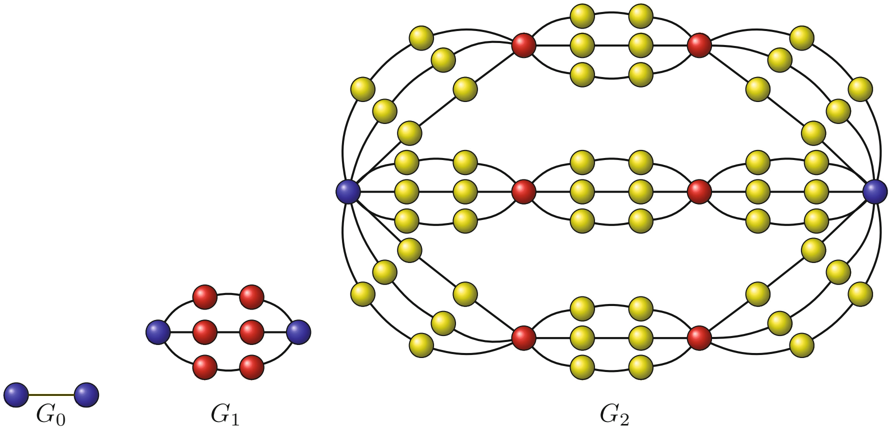

Fig. 4. The 3 iterations of the 3-Flower graph.

Using this construction, it is possible to give the exact values for the properties of the k-Flower graph. The number of edges of G_n is calculated as follows:

$$E_{G_n} = E_{h_k^n(G_0)} = E_{h_k(G_{n-1})} = E_{B_k \circ R_k(G_{n-1})}$$

Using Lemma 2, we obtain: $E_{G_n} = k \times E_{R_k(G_{n-1})}$

And using Lemma 4, we obtain:

$$E_{G_n} = k^2 \times E_{G_{n-1}} = k^4 \times E_{G_{n-2}} = k^6 \times E_{G_{n-3}} = ... = k^{2n} \times E_{G_0}$$

So the number of edges of G_n is $E_{G_n} = k^{2n}$

Similarly, we find:

The number of vertices of G_n**:** $V_{G_n} = 2 + \frac{k(k^{2n}-1)}{k+1}$

The number of faces of G_n: $F_{G_n} = 1 + \frac{k^{2n}-1}{k+1}$

The average degree of G_n: $< z >_{G_n} = \frac{2E_{G_n}}{V_{G_n}} = \frac{2k^{2n}}{2+\frac{k(k^{2n}-1)}{k+1}}$

4 The Number of Spanning Trees of a Flower Network

Due to the large size of self-similar networks, their complexity is very difficult to compute, even if we use the theorem of Kirchhoff. For this reason, we use these combinatorial approaches that facilitate this computation. We choose the Flower network as a self-similar network because of its interesting topological structure. Using the above theorems and the proposed methods in the last section, we obtain the exact number of spanning trees in the 2-Flower network and the k-Flower network:

4.1 The Number of Spanning Tree of a 2-Flower Graph:

We combine Our approaches: First, we apply the reduction approach, then the bipartite approach to calculate the number of spanning trees of the 2-Flower graph.

Theorem 5. *Let C_n denote a 2-Flower graph where n is the number of iterations. The number of spanning trees of C_n is given by the following formula:*

$$\tau(C_n) = 2^{2[\frac{2^{2n}-1}{3}]} \tag{5}$$

Proof: This process can be presented as:

$$\tau(C_n) = \tau(h_2^n(C_0)) = \tau(h_2(C_{n-1})) = \tau(B_2(R_2(C_{n-1}))).$$

Using Theorem 1, we obtain:

$$\tau(C_n) = 2^{F_{R_2(C_{n-1})}-1} \times \tau(R_2(C_{n-1})) \text{ with } F_{R_2(C_{n-1})} = F_{C_n}$$

Using Theorem 3, we obtain:

$$\tau(C_n) = 2^{F_{C_n}-1} \times 2^{V_{C_{n-1}}-1} \times \tau(C_{n-1}).$$

Using the properties of the 2-Flower graph in Sect. 3, we obtain:

$$\begin{aligned}
\tau(C_n) &= 2^{\frac{2^{2n}-1}{3}} \times 2^{1+\frac{2}{3}(2^{2(n-1)}-1)} \times \tau(C_{n-1}) \\
\tau(C_n) &= 2^{2^{2n-1}} \times \tau(C_{n-1}) \\
\tau(C_n) &= 2^{2^{2n-1}+2^{2n-3}} \times \tau(C_{n-2}) \\
\tau(C_n) &= 2^{2^{2n-1}+2^{2n-3}+2^{2n-5}} \times \tau(C_{n-3}) \\
&\vdots
\end{aligned}$$

$$\tau(C_n) = 2^{2^{2n-1}+2^{2n-3}+2^{2n-5}+\ldots+2^1} \times \tau(C_0) \ with \ \tau(C_0) = 1$$
$$\tau(C_n) = 2^{2[(2^2)^{n-1}+(2^2)^{n-2}+(2^2)^{n-3}+\ldots+(2^2)^0]}$$
$$\tau(C_n) = 2^{2[\frac{2^{2n}-1}{3}]}.$$

4.2 The Number of Spanning Tree of a k-Flower Graph:

We combine Our approaches: First, we apply the k-reduced approach, then the k-partite approach to evaluate the number of spanning trees of the k-Flower network.

Theorem 6. *Let G_n denote a k-Flower graph where n is the number of iterations. The number of spanning trees of G_n is given by the following formula:*

$$\tau(G_n) = k^{k[\frac{k^{2n}-1}{k^2-1}]} \tag{6}$$

Proof: This process can be presented as:

$$\tau(G_n) = \tau(h_k^n(G_0)) = \tau(h_k(G_{n-1})) = \tau(B_k(R_k(G_{n-1}))).$$

Using Theorem 2, we obtain:

$$\tau(G_n) = k^{F_{R_k(G_{n-1})}-1} \times \tau(R_k(G_{n-1})) \text{ with } F_{R_k(G_{n-1})} = F_{G_n}$$

Using Theorem 4, we obtain:

$$\tau(G_n) = k^{F_{G_n}-1} \times k^{V_{G_{n-1}}-1} \times \tau(G_{n-1}).$$

Using the properties of the k-Flower graph in Sect. 3, we obtain:

$$\tau(G_n) = k^{\frac{k^{2n}-1}{k+1}} \times k^{1+\frac{k(k^{2(n-1)}-1)}{k+1}} \times \tau(G_{n-1})$$
$$\tau(G_n) = k^{k^{2n-1}} \times \tau(G_{n-1})$$
$$\tau(G_n) = k^{k^{2n-1}+k^{2n-3}} \times \tau(G_{n-2})$$
$$\tau(G_n) = k^{k^{2n-1}+k^{2n-3}+k^{2n-5}} \times \tau(G_{n-3})$$
$$\vdots$$
$$\tau(G_n) = k^{k^{2n-1}+k^{2n-3}+k^{2n-5}+\ldots+k^1} \times \tau(G_0) \ with \ \tau(G_0) = 1$$
$$\tau(G_n) = k^{k[(k^2)^{n-1}+(k^2)^{n-2}+(k^2)^{n-3}+\ldots+(k^2)^0]}$$
$$\tau(G_n) = k^{k[\frac{k^{2n}-1}{k^2-1}]}.$$

Based on the Eqs. 5 and 6, we calculate the complexity of the k-Flower network based on its size:

Table 1. The numerical result of the complexity of the k-Flower network G_n.

k	n	$\tau(G_n)$
2	10	$1,0434 \times 10^{210435}$
3	7	$7,67435 \times 10^{855770}$
4	5	$1,0346 \times 10^{168348}$
5	4	$1,5099 \times 10^{56882}$
6	4	$1,5927 \times 10^{224056}$
7	4	$8,3606 \times 10^{710473}$

The numerical result: The below table presents some values of the number of spanning trees or the complexity of the k-Flower network G_n. These computations were performed using Maple by changing the value of the number of iterations n and the dimension of the Flower network k (Table 1).

5 The Spanning Tree Entropy in Flower Network

The entropy of spanning trees of a network or the asymptotic complexity is a quantitative measure of the number of spanning trees and it characterizes the network structure. We use this entropy to quantify the robustness of networks. The most robust network is the network that has the highest entropy. When the spanning trees number of networks grows exponentially with the number of vertices of G_n as $V_{G_n} \to \infty$, there exist a constant ρ_k describing this exponential growth, which is defined as:

$$\rho_k = \lim_{V_{G_n} \to \infty} \frac{\ln \tau(G_n)}{V_{G_n}}. \tag{7}$$

with $\tau(G_n)$ is the number of spanning trees of G_n and V_{G_n} is the number of vertices of G_n.

This formula provides the computation of the spanning tree entropy for the k-Flower graph.

Corollary 1. *The spanning trees entropy of the k-Flower graph is given by:*

$$\rho_k = \frac{ln(k)}{k-1} \tag{8}$$

Proof: We calculate the spanning trees entropy of a k-Flower network by: $\rho_k = \lim\limits_{V_{G_n} \to \infty} \frac{\ln|\tau(G_n)|}{|V_{G_n}|} = \lim\limits_{V_{G_n} \to \infty} \frac{\ln(k^{k[\frac{k^{2n}-1}{k^2-1}]})}{2+\frac{k(k^{2n}-1)}{k+1}} = \lim\limits_{V_{G_n} \to \infty} \frac{\ln(k)\times(k+1)}{k^2-1}$.

Then, the result. Similarly, we can find the entropy of the 2-Flower graph:

$$\rho_2 = ln2 = 0.6931$$

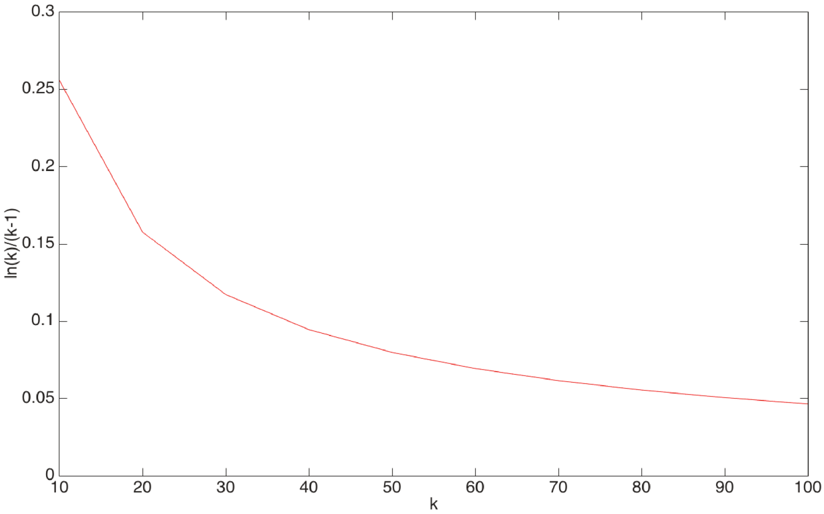

Fig. 5. The spanning tree entropy of the k-Flower graph

.

Knowing the number of spanning trees for the Flower network allows us to calculate its spanning tree entropy. In Fig. 5, we show that the entropy of spanning trees of the k-Flower graph varies with the dimension k and the increasing of this value leads to decrement the entropy of spanning trees. From this result, we deduce that the Flower networks with a higher dimension are less robust than those with a lower dimension.

6 Comparison with Other Networks Having the Same Average Degree

The value of ρ_2 is compared with the entropy of other networks having the same average degree (See the table below). The bigger the entropy value, the more the number of spanning trees, the network is more robust with the stronger heterogeneous topology because the increase of the number of spanning trees provides more possibilities of connecting two nodes related by defective links, that ensures a good reliability, robustness and availability of communication networks.

From the Table 2, we compare the entropy of the spanning trees of the 2-Flower network C_n with those of other networks with the same average degree. We notice the value of the entropy of the spanning trees of the 2-Flower network is the highest reported for the Koch networks and the Hanoi networks and it is the lowest reported for the 3-2-12 lattices, the 4-8-8 bathroom tile and Honeycomb lattice. This reflects the fact that the 2-Flower network has an average spanning tree rate compared to other networks with the same average degree.

Table 2. The entropy of several networks having the same average degree.

Type of network	$< z >$	ρ
Koch network [10]	3	0.549
Hanoi networks [11]	3	0.677
2-Flower networks	**3**	**0.6931**
The 3-2-12 lattices [12]	3	0.721
The 4-8-8 bathroom tile [12]	3	0.787
Honeycomb lattice [13]	3	0.807

This result proves that the 2-Flower network is more robust than the Koch networks and the Hanoi networks, on the other hand, the 2-Flower network is less robust than the 3-2-12 lattices, the 4-8-8 bathroom tile and Honeycomb lattice.

7 Conclusion

The concept of spanning tree entropy of a network is used to measure network robustness. In this paper, we have investigated a family of self-similar complex networks: A Flower network. We proposed two combinatorial approaches: The bipartition and the reduction to construct this network, to determine its topological properties and calculate its number of spanning trees. Finally, we evaluated its spanning tree entropy and compared it with those for other studied networks with the same average degree in order to estimate the most robust network between them.

References

1. Cook, A., Zanin, M.: Complex network theory. In: Complexity Science in Air Traffic Management, vol. 9 (2016)
2. Gao, J., et al.: Robustness of a network of networks. Phys. Rev. Lett. **107**(19), 195701 (2011)
3. Sorkhoh, I., Mahdi, K., Safar, M.: Cyclic entropy of complex networks. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE (2012)
4. Lyons, R.: Asymptotic enumeration of spanning trees. Comb. Probab. Comput. **14**(04), 491–522 (2005)
5. Wu, B.Y., Chao, K.-M.: Spanning Trees and Optimization Problems. CRC Press, Boca Raton (2004)
6. Kirchhoff, G.: Ueber die Auflsung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Strme gefhrt wird. Annalen der Physik **148**(12), 497–508 (1847)
7. Merris, R.: Laplacian matrices of graphs: a survey. Linear Algebra Appl. **197**, 143–176 (1994)
8. Lin, Y., et al.: Counting spanning trees in self-similar networks by evaluating determinants. J. Math. Phys. **52**(11), 113303 (2011)

9. Lotfi, D., Marraki, M.E., Aboutajdine, D.: The enumeration of spanning trees in dual, bipartite and reduced graphs. J. Discrete Math. Sci. Crypt. **18**(6), 673–687 (2015)
10. Zhang, Z., et al.: Mapping Koch curves into scale-free small-world networks. J. Phys. A Math. Theor. **43**(39), 395101 (2010)
11. Zhang, Z., et al.: The number and degree distribution of spanning trees in the Tower of Hanoi graph. Theor. Comput. Sci. **609**, 443–455 (2016)
12. Wu, F.Y.: Number of spanning trees on a lattice. J. Phys. A Math. Gen. **10**(6), L113 (1977)
13. Shrock, R., Fa Yueh, W.: Spanning trees on graphs and lattices in d dimensions. J. Phys. A Math. Gen. **33**(21), 3881 (2000)