axioms

MDPI

Article

A Method for Calculating the Reliability of 2-Separable Networks and Its Applications

Jing Liang [1,2], Haixing Zhao [3,4,*] and Sun Xie [5]

[1] School of Computer and Information Science, Qinghai Institute of Technology, Xining 810016, China; liangjingqh@163.com
[2] School of Computer Technology and Applications, Qinghai University, Xining 810016, China
[3] The State Key Laboratory of Tibetan Intelligent Information Processing and Application, Xining 810008, China
[4] Academy of Plateau, Science and Sustainability, Xining 810008, China
[5] School of Mathematics and Physics, Qinghai University, Xining 810016, China; ahwwxs@163.com
* Correspondence: h.x.zhao@163.com

Abstract: This paper proposes a computational method for the reliability of 2-separable networks. Based on graph theory and probability theory, this method simplifies the calculation process by constructing a network equivalent model and designing corresponding algorithms to achieve the efficient evaluation of reliability. Considering independent random failures of edges with equal probability q, this method can accurately calculate the reliability of 2-separable networks, and its effectiveness and accuracy are verified through examples. In addition, to demonstrate the generality of our method, we have also applied it to other 2-separable networks with fractal structures and proposed linear algorithms for calculating their all-terminal reliability.

Keywords: 2-separable network; fractal network; network reliability; the all-terminal reliability

MSC: 05C31; 68R10

check for updates

Citation: Liang, J.; Zhao, H.; Xie, S. A Method for Calculating the Reliability of 2-Separable Networks and Its Applications. *Axioms* **2024**, *13*, 459. https://doi.org/10.3390/axioms13070459

Academic Editor: Fevrier Valdez

Received: 19 April 2024
Revised: 16 June 2024
Accepted: 2 July 2024
Published: 8 July 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Network reliability is one of the hot topics in the fields of network performance analysis and combinatorial mathematics. Researchers often use graph theory models to delve into the reliability of networks. During the research process, researchers mainly classified network reliability into three types of models. The first type assumes that edges are perfectly reliable, while vertices fail independently with a fixed probability [1]. The second type of model assumes that vertices are perfectly reliable, while edges fail independently with a fixed probability [2,3]. The third type of model assumes that vertices and edges fail independently with a fixed probability [4,5]. The research on network reliability mainly focuses on reliability analysis and reliability design. Reliability analysis is the calculation of the reliability of a given network by a specified amount [6]. Reliability design refers to finding a graph with the maximum reliability polynomial or the minimum unreliability polynomial in a graph with the same number of vertices and edges [3,7].

In 1956, Moore and Shannon [8] introduced a probabilistic model firstly to study network reliability, in which vertices are considered to be perfectly reliable, and edges fail independently with a fixed probability. Within this model, there are some research branches. One of the research branches is the all-terminal reliability. For complex networks with edges that fail with a certain probability, the all-terminal reliability, which is the probability that the surviving edges induce a connected spanning subgraph, is an important evaluation index in system science to evaluate the survivability of the corresponding complex networks [9]. Suppose that $G = (V, E)$ is a graph with edges that fail independently with the identical probability q. The edge failure reliability polynomial can be expressed as $R_e(G, q) = \sum_{i=n-1}^{e} N_i(1-q)^i q^{e-i}$, where e is the number of edges of G, N_i denotes the number of connected spanning subgraphs with i edges of

G. Then, the disconnected probability $P_e(G,q)$ of G is defined as $P_e(G,q) = \sum_{i=1}^{e} m_i q^i (1-q)^{e-i}$, where m_i denotes the number of cut-sets with i edges of G. If $0 \leq q \leq 1$ and all the graphs G' with the same number of nodes and edges as G have $Re(G,q) \geq Re(G',q)$, then the graph G is uniformly optimal under edge failure. If $0 \leq q \leq 1$ and all the graphs G' with the same number of nodes and edges as G have $Re(G,q) \leq Re(G',q)$, then the graph G is uniformly worst under edge failure.

It is NP-hard to compute $R_e(G,q)$ for general networks G [10]. Ball and Provan in [11,12] proved that calculating the all-terminal reliability of a graph, even if for planar graphs, is ♯p-complete. Therefore, researchers have been looking for polynomial-time algorithms for networks with special structures. Up to now, linear-time algorithms [10,13,14] have been developed for some symmetric networks, such as the SP (series–parallel) network, PCF (Planar–Cube–Free) network, and planar 2-tree network.

The rest of this paper is organized as follows. Section 2 introduces a new method for calculating the all-terminal reliability of 2-separable networks. In Section 3, based on the new method in Section 2, two linear algorithms are proposed for the all-terminal reliability in two types of fractal networks, such as $H(K_{2,n}, t)$ and (x,y)-flower. Then, the uniformly optimal structure and uniformly worst structure of the (x, y)-flower network can be determined by the algorithm. The research results of this paper are summarized in Section 4.

2. Preliminaries

A network with n vertices and m edges can be modeled as a graph G with the same number of vertices, edges, and interconnections as the network. For example, in a transportation network, stations are represented as nodes, and if there is a link between two stations, the corresponding nodes are connected to each other with edges, thus obtaining the corresponding graph structure. Later, the concepts of a network and graph are no longer distinguished. All graphs considered here are simple and undirected. A graph G is said to be connected if each pair of different vertices u, v has a (u, v) path. Let $G = (V(G), E(G))$, where a subgraph H satisfies $V(H) = V(G)$, and $E(H) \subseteq E(G)$ is called a spanning subgraph of G. A complete graph is a simple graph where any two of its vertices are adjacent, which is denoted by K_n with n vertices. A path with n vertices is denoted by P_n, and a cycle on n vertices is denoted by C_n. A graph is bipartite if its vertex set can be partitioned into two subsets X and Y so that every edge has one end in X and the other end in Y; such a partition (X,Y) is called a bipartition of the graph, and X and Y are its parts. We denote a bipartite graph G with bipartition (X,Y) by $G[X, Y]$. If $G[X, Y]$ is simple and every vertex in X is linked to every vertex in Y, then G is called a complete bipartite graph. Let $\Omega(n, m)$ denote the set of all connected graphs with n vertices and m edges, with no loops or multiple edges. For other standard concepts and terminology, please refer to [15].

A fractal is a type of shape with self-similar structures, meaning that the structure remains the same at different magnifications [16,17]. Therefore, it can be simply understood as follows: a fractal object is divisible into several parts, each of which is a reduction in the overall shape. Therefore, fractal networks have basic properties such as self-similarity, scale-free, and self-affine. Common fractal networks include fractal scale-free networks [18], sierpinski gasket [19], pseudo fractal scale-free networks [20], etc. There have been some research results on these networks and their related applications [21].

In this paper, we focus on 2-separable network models [22,23] and study the computation method and generalization of their all-terminal reliability. This research area is of great relevance due to the 2-connectivity characteristics exhibited by numerous networks in practical applications [14,24,25]. The definition of a 2-separable network is given in [26].

For $1 \leq i \leq m$, let $G = (V(G), E(G))$ be a network that has m edges $e_i = (a_i, b_i)$, and $H_{a_i}^{b_i}$ be m connected networks, where a_i and b_i are two end nodes of $H_{a_i}^{b_i}$. In using $H_{a_i}^{b_i}$ to replace each edge of G acording to the end nodes, the resulting graph formed is a 2-separable graph (see Figure 1), which is denoted by $G(H_{a_i}^{b_i}|_1^m)$.

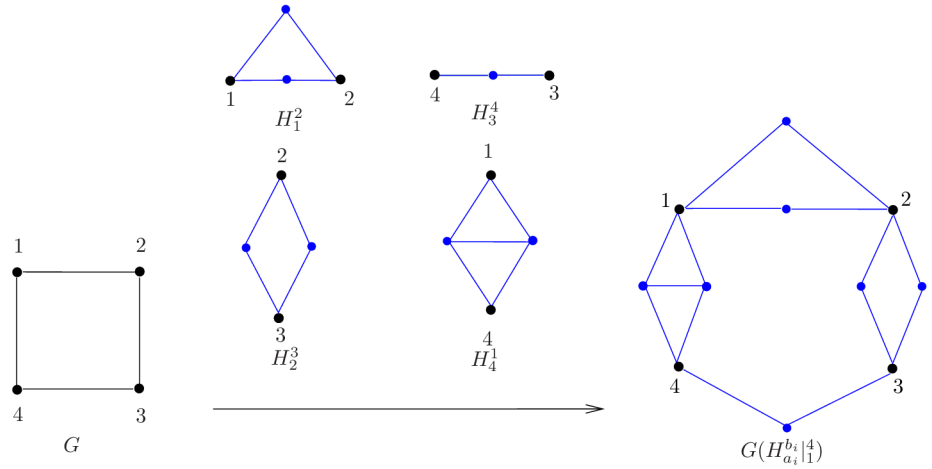


Figure 1. The formation process of 2-separable network $G(H_{a_i}^{b_i}|_1^4)$.

The graphs generated by replacements have received attention in several aspects, including but not limited to perfect matching [27], the counting of spanning trees and spanning forests [26,28,29], the resistance distance [30], the matching polynomial [31], and the Tutte polynomial [23,32]. Until now, there have been few results for calculating the all-terminal reliability of 2-separable networks. In this paper, according to the formation characteristics of 2-separable network, we present a new method for calculating its all-terminal reliability. Based on this method, two linear time algorithms are proposed to calculate the all-terminal reliability of two self-similar networks. The time complexity of the algorithm is O(t), where t is the time steps. For a given network, the number of spanning trees is a key parameter used to measure its reliability in edge failure cases. However, using the method of this paper, the number of spanning trees of two fractal networks can be acquired in linear time.

3. Reliability of 2-Separable Network

In this section, a new method for calculating the all-terminal reliability of a 2-separable network is proposed. Each edge of a graph G is weighted by a pair of real numbers (p, q), where p and q denote the reliability and failure probability of the edge, respectively. In order to facilitate the description in this paper, $R_e(G, p, q)$ is used to represent the all-terminal reliability of network G when the edge reliability probability is p and the edge failure probability is q.

Theorem 1. *Let $G(H_{a_i}^{b_i}|_m^1)$ be a 2-separable network and $H_{a_i}^{b_i}(1 \le i \le m)$ be m connected networks. Let $\alpha_{a_i}^{b_i}$ be the connected probability of a_i and b_i in $H_{a_i}^{b_i}$, and let $\beta_{a_i}^{b_i}$ be the disconnected probability of a_i and b_i in $H_{a_i}^{b_i}$, where there are just two components. Therefore, the all-terminal reliability of $G(H_{a_i}^{b_i}|_m^1)$ is*

$$R_e(G(H_{a_i}^{b_i}|_m^1), p, q) = R_e(G, \alpha_{e_i}, \beta_{e_i})$$

where p and q denote the edge reliability probability and edge failure probability of the network $G(H_{a_i}^{b_i}|_m^1)$, respectively; α_{e_i} and β_{e_i} denote the edge reliability probability and edge failure probability of e_i in G, which is the graph obtained from $G(H_{a_i}^{b_i}|_m^1)$ by replacing $H_{a_i}^{b_i}$ with $e_i = (a_i, b_i)$.

Proof of Theorem 1. Suppose that $G(H_{a_i}^{b_i}|_m^1)$ is a 2-separable network and $H_{a_i}^{b_i}(1 \le i \le m)$ is m connected networks. Let $\alpha_{a_i}^{b_i}$ be the connected probability of a_i and b_i in $H_{a_i}^{b_i}$ and $\beta_{a_i}^{b_i}$ be the disconnected probability of a_i and b_i in $H_{a_i}^{b_i}$, where there are just two components.

In order to calculate the all-terminal reliability of 2-separable network $G(H_{a_i}^{b_i}|_m^1)$, we need to focus our attention on the construction of 2-separable network $G(H_{a_i}^{b_i}|_m^1)$. Trying to utilize the all-terminal reliability of G, one can obtain the all-terminal reliability of

$G(H_{a_i}^{b_i}|_m^1)$. It is important for us to obtain G. The graph G is achieved by replacing $H_{a_i}^{b_i}$ with $e_i = (a_i, b_i)$. Edges in G are weighted by α_{e_i} and β_{e_i}, which are the edge reliability probability and edge failure probability, respectively. The edge reliability probability α_{e_i} of e_i equals the connected probability of vertices a_i and b_i in $H_{a_i}^{b_i}$, which means that $\alpha_{e_i} = \alpha_{a_i}^{b_i}$; the edge failure probability β_{e_i} of e_i equals the disconnected probability of vertices a_i and b_i in $H_{a_i}^{b_i}$ with just two components, which means that $\beta_{e_i} = \beta_{a_i}^{b_i}$. The all-terminal reliability of network G is $R_e(G, \alpha_{e_i}, \beta_{e_i})$. From this, we can obtain $R_e(G(H_{a_i}^{b_i}|_m^1), p, q) = R_e(G, \alpha_{e_i}, \beta_{e_i})$. The proof is completed. □

Example 1. *The all-terminal reliability of $G(H_{a_i}^{b_i}|_4^1)$, which is shown in Figure 2, can be calculated by Theorem 1. The specific analysis is as follows.*

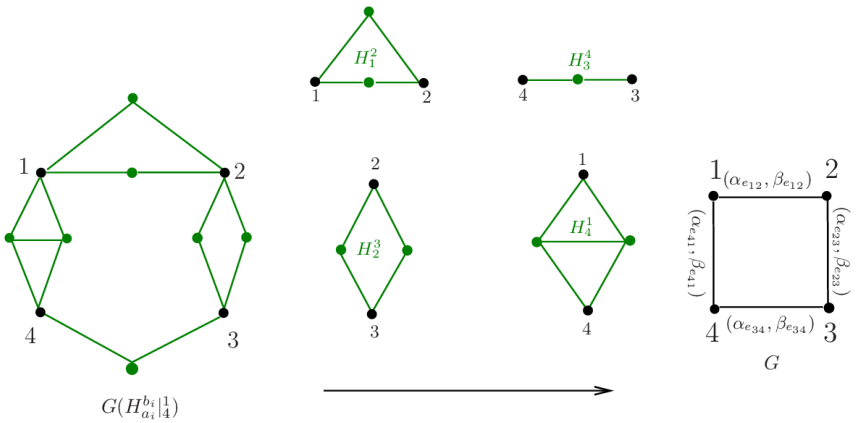


Figure 2. The reduction process of network $G(H_{a_i}^{b_i}|_4^1)$ to G.

Suppose that $G(H_{a_i}^{b_i}|_4^1)$ is a 2-separable network. Set $a_1 = b_4 = 1$, $a_2 = b_1 = 2$, $a_3 = b_2 = 3$, and $a_4 = b_3 = 4$. Then, $H_{a_1}^{b_1}$, $H_{a_2}^{b_2}$, $H_{a_3}^{b_3}$, and $H_{a_4}^{b_4}$ can be substituted by H_1^2, H_2^3, H_3^4, and H_4^1, which are connected. Let α_i^j be the connected probability of vertices i and j in H_i^j and β_i^j be the disconnected probability of vertices i and j in H_i^j, where there are just two components.

By replacing H_1^2, H_2^3, H_3^4, and H_4^1 with $e_1 = (1,2)$, $e_2 = (2,3)$, $e_3 = (3,4)$, and $e_4 = (4,1)$, one can obtain a simple graph G, with vertex set $V(G) = \{1,2,3,4\}$ and edge set $E(G) = \{e_1, e_2, e_3, e_4\}$. For each edge of G, the reliability probability and the failure probability are as follows:

$$\begin{aligned}
&\alpha_{e_1} = \alpha_1^2 = 4p^3q + p^4, && \beta_{e_1} = \beta_1^2 = 4p^2q^2;\\
&\alpha_{e_2} = \alpha_2^3 = 4p^3q + p^4, && \beta_{e_2} = \beta_2^3 = 4p^2q^2;\\
&\alpha_{e_3} = \alpha_3^4 = p^2, && \beta_{e_3} = \beta_3^4 = 2pq;\\
&\alpha_{e_4} = \alpha_4^1 = 8p^3q^2 + 5p^4q + p^5, && \beta_{e_4} = \beta_4^1 = 8p^2q^3 + 2p^3q^2.
\end{aligned}$$

According to Theorem 1, it is obtained that

$$\begin{aligned}
R_e(G(H_{a_i}^{b_i}|_4^1), p, q) = R_e(G, \alpha_{e_i}, \beta_{e_i}) = \beta_{e_1}\alpha_{e_2}\alpha_{e_3}\alpha_{e_4} + \alpha_{e_1}\beta_{e_2}\alpha_{e_3}\alpha_{e_4} + \alpha_{e_1}\alpha_{e_2}\beta_{e_3}\alpha_{e_4}\\
+\alpha_{e_1}\alpha_{e_2}\alpha_{e_3}\beta_{e_4} + \alpha_{e_1}\alpha_{e_2}\alpha_{e_3}\alpha_{e_4} = 640p^{10}q^5 + 736p^{11}q^4 + 368p^{12}q^3 + 100p^{13}q^2 + 15p^{14}q + p^{15}.
\end{aligned}$$

Moreover, the all-terminal reliability of 2-separable network $G(H_{a_i}^{b_i}|_4^1)$ was calculated by MATLAB R2019b directly: $R_e(G(H_{a_i}^{b_i}|_4^1), p, q) = 640p^{10}q^5 + 736p^{11}q^4 + 368p^{12}q^3 + 100p^{13}q^2 + 15p^{14}q + p^{15}$, which indicates that the method proposed in this paper is effective. For detailed computational ideas, please see Algorithm A1 in Appendix A, which is a pseudo-code for computing the coefficients N_{n-3} of the reliability polynomials of the all-terminal network by MATLAB R2019b.

4. Applications

This section mainly presents the application of Theorem 1. We propose two linear algorithms to calculate the all-terminal reliability of two types of networks with fractal structure.

4.1. Hierarchical Lattices

This subsection first introduces the model construction of hierarchical lattices with fractal scale-free characteristics and then proposes a linear algorithm to calculate the reliability of such networks.

4.1.1. Modeling Construction

The hierarchical lattice [33] is a type of fractal scale-free network constructed in an iterative manner, as shown in Figure 3. The network generated by t-step iteration is denoted as $H(K_{2,l}, t)$. The $K_{2,l}$ denotes connected clusters, and its node set can be divided into two subsets V_1 containing two nodes and V_2 containing l nodes.

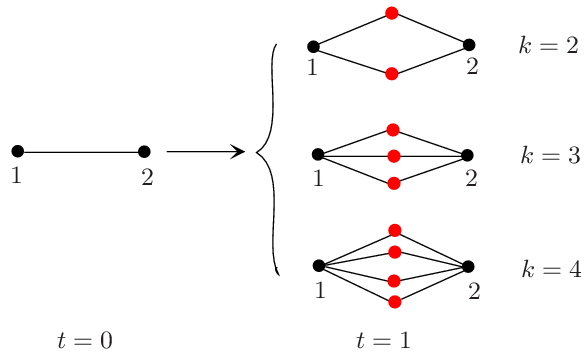


Figure 3. Several iterative construction methods of special hierarchical lattice.

Recall that $H(K_{2,l}, t)$ is obtained by following construction method. Using N_t and E_t denote the number of nodes and edges of $H(K_{2,l}, t)$, respectively.

Step 1. For $t = 0$, the initial network $H(K_{2,l}, 0)$ is an edge connecting two nodes.
Step 2. For $t = 1$, $H(K_{2,l}, 1)$ is derived from $H(K_{2,l}, 0)$. After deleting the edge of $H(K_{2,l}, 0)$, retaining two nodes of $H(K_{2,l}, 0)$ as V_1 of $H(k, 1)$, and adding l new nodes as V_2 of $H(K_{2,l}, 1)$, the node set of $H(K_{2,l}, 1)$ is the union of V_1 and V_2. The edges of $H(K_{2,l}, 1)$ are joining every node in V_1 to every node in V_2. In other words, $H(K_{2,l}, 1)$ is obtained using $K_{2,l}$ to substitute for $H(K_{2,l}, 0)$.
Step 3. For $t > 1$, $H(K_{2,l}, t)$ is an evolution of $H(K_{2,l}, t-1)$. $K_{2,l}$ substitutes for each edge of $H(K_{2,l}, t-1)$ in the same way as $K_{2,l}$ instead of $H(K_{2,l}, 0)$.

After t steps, it is easy to calculate N_t and E_t, where $N_t = \frac{l\cdot(2l)^t+3l-2}{2l-1}$ and $E_t = (2l)^t$. Figure 4 shows the first three steps of the iterative process of $H(K_{2,l}, 2)$.

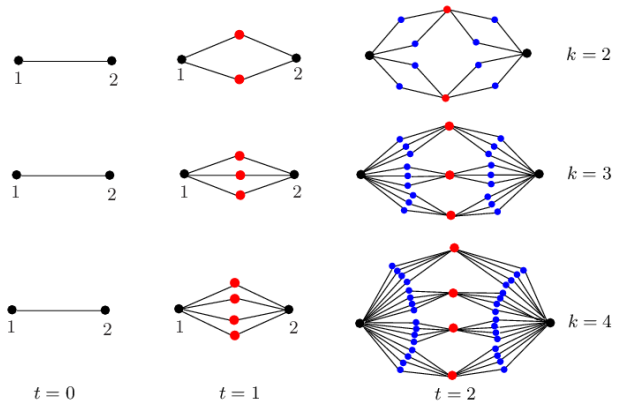


Figure 4. The first three iterations of hierarchical lattices in three special cases.

4.1.2. Algorithm Description

According to the definition of a 2-separable network, $H(K_{2,l}, t)$ is a special 2-separable network, where $G = K_2$, $H_{a_i}^{b_i} = K_{2,l}$, and $H(K_{2,l}, t) = H(K_{2,l}, t-1)[K_{2,l}]$. Theorem 1 provides a useful method for calculating the all-terminal reliability of 2-separable networks. Based on the method, a linear algorithm is raised to compute the all-terminal reliability of $H(K_{2,l}, t)$.

Algorithm 1 Initial condition: Let $H(K_{2,l}, t)$ be a hierarchical lattice generated by t-step iterations. Let $R_e(H(K_{2,l}, t), p, q)$ be the all-terminal reliability of $H(K_{2,l}, t)$. In $H(K_{2,l}, t)$, each edge is weighted by a pair of real numbers (p, q), where p and q denote the edge reliability probability and edge failure probability, respectively.

Step 1. Take $i = t$.

Step 2. The weight of each edge in $H(K_{2,l}, i)$ is denoted by $(\alpha_{t-i}, \beta_{t-i})$. If there is a subgraph isomorphic to $K_{2,l}$ in $H(K_{2,l}, i)$, then replace every such subgraph by an edge with endpoints that are the vertices in the vertex subset V_1 of $K_{2,l}$ and obtain $H(K_{2,l}, i-1)$ with $(\alpha_{t+1-i}, \beta_{t+1-i})$, where

$$\begin{cases} \alpha_{t+1-i} := 2^{l-1} l \alpha_{t-i}^{l+1} \beta_{t-i}^{l-1} + 2^{l-2} l \alpha_{t-i}^{l+2} \beta_{t-i}^{l-2} + \cdots + 2l \alpha_{t-i}^{2l-1} \beta_{t-i} + \alpha_{t-i}^{2l}, \\ \beta_{t+1-i} := 2^l \alpha_{t-i}^l \beta_{t-i}^l. \end{cases} \tag{1}$$

Step 3. If $i = 0$, the algorithm is stopped, and $R_e(H(K_{2,l}, t), p, q) := \alpha_t$ is output. Otherwise, set $i := i - 1$ and go to Step 2.

It can be easily verified that the complexity of Algorithm 1 is O(t), where t is the time steps. The pseudo-code of Algorithm 1 is given below.

Algorithm 1: The calculation of all-terminal reliability of network $H(K_{2,l}, t)$

Input: l, t
Output: $R_e(H(K_{2,l}, t), p, q)$
1 $\alpha_0 = p$, $\beta_0 = q$;
2 **for** $i \leftarrow t$ *to* 1 **do**
3 % $K_{2,l}^i$ is a bipartite graph containing l vertices of degree 2 in $H(K_{2,l}, i)$;
4 replace every $K_{2,l}$ by an edge with endpoints that are the vertices in the vertex subset V_1 of $K_{2,l}$ and obtain $H(K_{2,l}, i-1)$ with $(\alpha_{t+1-i}, \beta_{t+1-i})$;
5 $\alpha_{t+1-i} = 2^{l-1} l \alpha_{t-i}^{l+1} \beta_{t-i}^{l-1} + 2^{l-2} l \alpha_{t-i}^{l+2} \beta_{t-i}^{l-2} + \cdots + 2l \alpha_{t-i}^{2l-1} \beta_{t-i} + \alpha_{t-i}^{2l}$;
6 $\beta_{t+1-i} = 2^l \alpha_{t-i}^l \beta_{t-i}^l$;
7 **end**
8 Output α_t, that is, $R_e(H(K_{2,l}, t), p, q)$.

Through Algorithm 1, the coefficients of the reliability polynomial of $K_{2,l}$ can be obtained in the following Table 1,

Table 1. The all-terminal reliability of $K_{2,l}$.

Network	Reliability Polynomial Coefficient						
$K_{2,2}$	2^0	$2^1 \times 2$					
$K_{2,3}$	2^0	$2^1 \times 3$	$2^2 \times 3$				
$K_{2,4}$	2^0	$2^1 \times 4$	$2^2 \times 4$	$2^3 \times 4$			
...	...	...	...	...	...		
$K_{2,l-1}$	2^0	$2^1 \times (l-1)$	$2^2 \times (l-1)$	...	...	$2^l \times (l-1)$	
$K_{2,l}$	2^0	$2^1 \times l$	$2^2 \times l$	...	...	...	$2^{l-1} \times l$

According to Table 1, the all-terminal reliability polynomial of $K_{2,l}$ is

$$R_e(K_{2,l}, p, q) = p^{2l} + 2lp^{2l-1}q + 4lp^{2l-2}q^2 + \cdots + 2^{l-2} l p^{l+2} q^{l-2} + 2^{l-1} l p^{l+1} q^{l-1}.$$

Example 2. *The all-terminal reliability of $H(K_{2,3},2)$, which is shown in Figure 5, can be calculated by Algorithm 1. The specific analysis is as follows.*

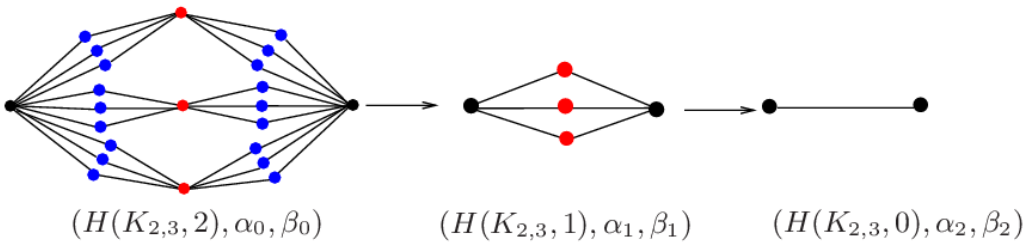


Figure 5. The calculation process of $R_e(H(K_{2,3},2),p,q)$.

Since $H(K_{2,3},2)$ is a special 2-separable network, in using Algorithm 1, the following can be obtained:

$$\begin{cases} \alpha_0 = p \\ \beta_0 = q \end{cases} \Rightarrow \begin{cases} \alpha_1 = 12\alpha_0^4\beta_0^2 + 6\alpha_0^5\beta_0 + \alpha_0^6 \\ \beta_1 = 8\alpha_0^3\beta_0^3 \end{cases} \Rightarrow \begin{cases} \alpha_2 = 12\alpha_1^4\beta_1^2 + 6\alpha_1^5\beta_1 + \alpha_1^6 \\ \beta_2 = 8\alpha_1^3\beta_1^3 \end{cases}$$

Because $R_e(H(K_{2,3},2),p,q) = \alpha_2$,

$$\begin{aligned} &R_e(SN(K_{2,3})_2,p,q) = 12(12\alpha_0^4\beta_0^2 + 6\alpha_0^5\beta_0 + \alpha_0^6)^4(8\alpha_0^3\beta_0^3)^2 + 6(12\alpha_0^4\beta_0^2 + 6\alpha_0^5\beta_0 + \alpha_0^6)^5 \\ &(8\alpha_0^3\beta_0^3) + (12\alpha_0^4\beta_0^2 + 6\alpha_0^5\beta_0 + \alpha_0^6)^6 \\ &= p^{36} + 36p^{35}q + 612p^{34}q^2 + 6528p^{33}q^3 + 48960p^{32}q^4 + 274176p^{31}q^5 + 1187904p^{30}q^6 \\ &+ 4068864p^{29}q^7 + 11151360p^{28}q^8 + 24551424p^{27}q^9 + 43213824p^{26}q^{10} + 59719680p^{25}q^{11} \\ &+ 62042112p^{24}q^{12} + 43794432p^{23}q^{13} + 15925248p^{22}q^{14}. \end{aligned}$$

In addition, the MATLAB R2019b program can be run to obtain the same $R_e(H(K_{2,3},2),p,q)$ as Algorithm 1, which proves the correctness and effectiveness of Algorithm 1. However, if the network is large, the computational complexity of MATLAB R2019b is very high, and Algorithm 1 can be obtained in linear time.

4.2. The (x, y)-Flower Network

The (x, y)-flower networks have the same degree sequences and grow according to a deterministic model with time evolution. The iterative construction process of (x, y)-flower networks can be found in [34,35].

4.2.1. Modeling Construction

In this paper, let x and y be positive integers, $1 \leq x \leq y$ and $x + y \geq 3$. Let $F_t(x,y)(t > 0)$ be the (x, y)-flower network generated by iterating over the following t steps:

Step 1. For $t = 0$, $F_0(x,y)$ has only one edge connecting two vertices, which is K_2.

Step 2. For $t = 1$, $F_1(x,y)$ is derived from $F_0(x,y)$. The edge in $F_0(x,y)$ will be replaced by two parallel paths of length x and y. In other words, one path, called the x path, contains $x - 1$ new vertices and two old vertices, which are in $F_0(x,y)$, and the other path, called the y path, contains $y - 1$ new vertices and two old vertices in $F_0(x,y)$.

Step 3. For $t > 1$, $F_t(x,y)$ is obtained from $F_{t-1}(x,y)$. Each edge in $F_{t-1}(x,y)$ will be replaced by two newly parallel paths with length x and y, which are generated in the same manner as step 2.

Let N_t and M_t be the numbers of vertices and edges in the network $F_t(x,y)$, respectively. Obviously, $N_t = \frac{x+y-2}{x+y-1}(x+y)^t + \frac{x+y}{x+y-1}$, and $M_t = (x+y)^t$.

Figures 6 and 7 show the iterative generation process of the first three steps of a (1,4)-flower and (2,3)-flower, respectively.

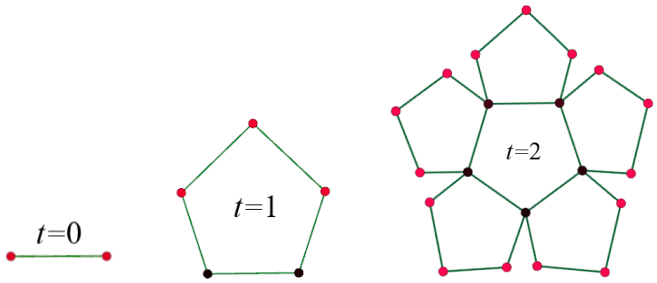


Figure 6. Generation process of (1 , 4)-flower.

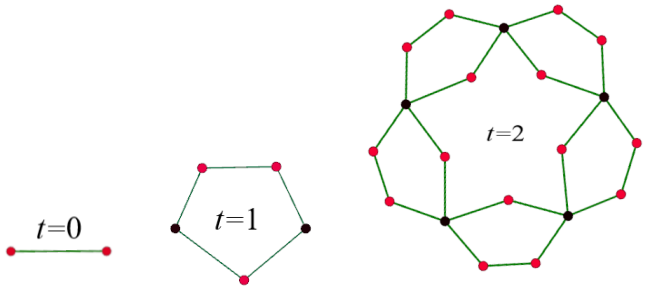


Figure 7. Generation process of (2 , 3)-flower.

4.2.2. Algorithm Description

For $t = 0$, the (x, y)-flower network is isomorphic to K_2, denoted by $F_0(x, y)$. Then, the all-terminal reliability of $F_0(x, y)$ is p. In the following, the (x, y)-flower network starts from $t = 1$. In combining the structural characteristics of the (x, y)-flower network and Theorem 1, a linear algorithm is proposed to calculate $R_e(F_t(x, y), p, q)$.

Algorithm 2: Initial condition: Let $F_t(x, y)$ be a (x, y)-flower network generated by t-step iterations. Let $R_e(F_t(x, y), p, q)$ be the all-terminal reliability of $F_t(x, y)$. Each edge is weighted by a pair of real numbers (p, q) in $F_t(x, y)$, where p and q denote the edge reliability probability and edge failure probability, respectively.

Step 1. Start from $i = t$.

Step 2. The weight of each edge in $F_t(x, y)$ is denoted by $(\alpha_{t+1-i}, \beta_{t+1-i})$. If there is a subgraph isomorphic to $C_{x,y}$ in $F_i(x, y)$, then replace every such subgraph by an edge with endpoints that have a vertex degree more than 2 for $C_{x,y}$ and obtain $F_{i-1}(x, y)$ with $(\alpha_{t+2-i}, \beta_{t+2-i})$, where

$$\begin{cases} \alpha_{t+2-i} := (x+y)\alpha_{t+1-i}^{x+y-1}\beta_{t+1-i} + \alpha_{t+1-i}^{x+y}, \\ \beta_{t+2-i} := xy\alpha_{t+1-i}^{x+y-2}\beta_{t+1-i}^{2}. \end{cases} \tag{2}$$

Step 3. If $i = 1$, the algorithm is stopped, and $R_e(F_t(x, y), p, q) := R_e(F_1(x, y), \alpha_t, \beta_t)$ is output. Otherwise, set $i := i - 1$ and go to Step 2.

It can be easily verified that the complexity of Algorithm 2 is O(t), where t is the time steps. The pseudo-code of Algorithm 2 is given below.

Algorithm 2: The calculation of all-terminal reliability of network $F_t(x,y)$

Input: a, t
Output: $R_e(F_t(x,y),p,q)$
Initialize $\alpha_0 = p$, $\beta_0 = q$;
for $x \leftarrow 1$ *to* $\lfloor \frac{a}{2} \rfloor$ **do**
  $y = a - x$;
  for $i \leftarrow t$ *to* 2 **do**
    % $C^i_{x,y}$ is a cycle containing $x+y-2$ vertices of degree 2 in $F_i(x,y)$;
    replace every $C^i_{x,y}$ by an edge with endpoints that have a vertex degree more than 2 for $C_{x,y}$ to obtain $F_{i-1}(x,y)$;
    $\alpha_{t+2-i} = (x+y)\alpha^{x+y-1}_{t+1-i}\beta_{t+1-i} + \alpha^{x+y}_{t+1-i}$;
    $\beta_{t+2-i} = xy\alpha^{x+y-2}_{t+1-i}\beta^2_{t+1-i}$;
  end
end
$R_e(F_t(x,y),p,q) = R_e(F_1(x,y),\alpha_t,\beta_t)$.

By Algorithm 2, the all-terminal reliability of $F_t(x,y)$ can be calculated by the all-terminal reliability of $F_1(x,y)$, i.e., $R_e(F_t(x,y),p,q) = R_e(F_1(x,y),\alpha_t,\beta_t)$. In the (x, y)-flower networks with special symmetric structure, the optimal and worst structures can be obtained by the following theorem.

4.2.3. Uniformly Optimal Network and Uniformly Worst Network of (x, y)-Flower

This section determines the uniformly optimal and uniformly worst structures of (x, y)-flower networks; see Theorem 2 for details. And this section verifies the results of Theorem 2 with a simple example.

Theorem 2. *Let x and y be positive integers and $1 \le x \le y$, $x + y = c$. Let $F_t(x,y)$ be the (x, y)-flower network with N_t vertices and M_t edges at time t. If $x = 1$, $F_t(1, c-1)$ is the uniformly worst network. If $x + y \equiv 0 (mod\ 2)$, then $F_t(\frac{c}{2}, \frac{c}{2})$ is the uniformly optimal network. If $x + y \equiv 1 (mod\ 2)$, then $F_t(\frac{c}{2}, \frac{c}{2}+1)$ is the uniformly optimal network.*

Proof of Theorem 2. In the (x, y)-flower network, $x + y$ is a constant denoted by c, i.e., $x + y = c$. From Algorithm 2, we can obtain that in the (x, y)-flower network, $R_e(F_t(x,y),\alpha_t,\beta_t) = R_e(F_1(x,y),\alpha_1,\beta_1)$. Therefore, the all-terminal reliability of $F_t(x,y)$ is related to α_t and β_t. By Algorithm 2, we can obtain $\alpha_{t+2-i} = (x+y)(\alpha_{t+1-i})^{x+y-1}\beta_{t+1-i} + (\alpha_{t+1-i})^{x+y}$ and $\beta_{t+2-i} = xy(\alpha_{t+1-i})^{x+y-2}\beta^2_{t+1-i}$. When α_{t+1-i} and β_{t+1-i} are determined, α_{t+2-i} can be obtained directly, but the value of β_{t+2-i} is determined by $x(c-x)$. Next, consider the value of $x(c-x)$. When $x = 1$ and $y = c-1$, $c-1$ is the minimum. At this time, $F_t(1, c-1)$ is the uniformly worst network. When $c \equiv 0 (mod\ 2)$, if $x = \frac{c}{2}$, then $x(c-x)$ is the maximum, and then β_{t+2-i} is the maximum. By iterating step by step, the largest α_t and β_t can be obtained. At this time, $F_t(\frac{c}{2}, \frac{c}{2})$ is a uniformly optimal network. When $c \equiv 1 (mod\ 2)$, if $x = \frac{c}{2}+1$, then $x(c-x)$ is the maximum, and then β_{t+2-i} is the maximum. By iterating step by step, the largest α_t and β_t can be obtained. At this time, $F_t(\frac{c}{2}, \frac{c}{2}+1)$ is a uniformly optimal network. □

Example 3. *The (x, y)-flower network at time t is $F_t(x,y)$. If $x + y = 6$ and $1 \le x \le y$, then there are three kinds of networks, namely $F_t(1,5)$, $F_t(2,4)$, and $F_t(3,3)$. The optimal structure of $F_t(x,y)$ can be built using Theorem 2. Take the reliability of these three networks at time $t = 2$ as an example, as shown in Figure 8. Consider the following:*

(1) $x = 1$, $y = 5$, $xy = 5$, the $(1,5)$-flower network is denoted by $F_t(1,5)$;
(2) $x = 2$, $y = 4$, $xy = 8$, the $(2,4)$-flower network is denoted by $F_t(2,4)$;
(3) $x = 3$, $y = 3$, $xy = 9$, the $(3,3)$-flower network is denoted by $F_t(3,3)$.

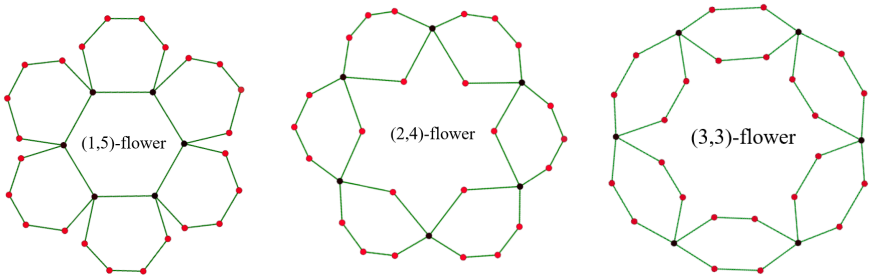


Figure 8. When $t = 2$, $F_2(1,5)$, $F_2(2,4)$ and $F_2(3,3)$.

Consequently, by comparing the reliability of the three networks at $t = 2$, the (x, y)-flower network $F_2(x, y)$ with an optimal structure is obtained as follows:

$$R_e(F_2(1,5),p,q) = R_e(F_1(1,5),\alpha_1,\beta_1) = 6\alpha_1^5\beta_1 + \alpha_1^6 = 6(6p^5q + p^6)^5(5p^4q^2) + (6p^5q + p^6)^6.$$
$$R_e(F_2(2,4),p,q) = R_e(F_1(2,4),\alpha_1,\beta_1) = 6\alpha_1^5\beta_1 + \alpha_1^6 = 6(6p^5q + p^6)^5(8p^4q^2) + (6p^5q + p^6)^6.$$
$$R_e(F_2(3,3),p,q) = R_e(F_1(3,3),\alpha_1,\beta_1) = 6\alpha_1^5\beta_1 + \alpha_1^6 = 6(6p^5q + p^6)^5(9p^4q^2) + (6p^5q + p^6)^6.$$

Therefore, when $t = 2$, the relationship between the all-terminal reliability of the above three networks, $F_2(1,5)$, $F_2(2,4)$, and $F_2(3,3)$, are ordered as follows: $R_e(F_2(3,3),p,q) > R_e(F_2(2,4),p,q) > R_e(F_2(1,5),p,q)$, which is shown in Figure 9 based on the MATLAB R2019b data simulation.

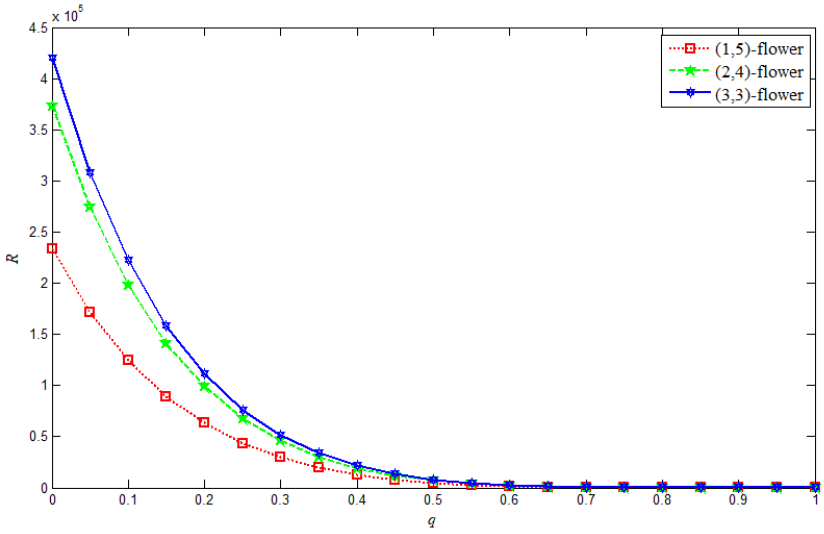


Figure 9. The relationship of $R_e(F_2(3,3),p,q)$, $R_e(F_2(2,4),p,q)$ and $R_e(F_2(1,5),p,q)$.

5. Conclusions

This paper proposes a new method for calculating the all-terminal reliability of 2-separable networks and has been effectively applied in specific scenarios of fractal structured networks. In network reliability analysis, the all-terminal reliability is an important indicator of network performance, which reflects the ability of the network to maintain connectivity even in the event of a failure. Based on the characteristics of 2-separable networks, this method uses a linear algorithm through deeply analyzing the structure and connection relationships of the network, which significantly improves the computational efficiency while maintaining computational accuracy. Especially when dealing with networks with fractal structures, this method can fully utilize the self-similarity of fractal networks, further simplify the calculation process, and achieve a fast calculation of reliability. The proposal of this new method not only enriches the theoretical system of network reliability analysis but also provides strong technical support for actual network design and maintenance.

Author Contributions: Conceptualization, methodology, software, validation, writing—original draft preparation, and writing—review and editing, J.L.; methodology and project administration, H.Z.; supervision, S.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the "Kunlun Talent" Talent Introduction Research Project (Grant No. 2023-QLGKLYCZX-027) and A Study of Reliability Indicators for Complex Networks and Their Applications (Grant No. 2023021wys019).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

This Appendix mainly provides MATLAB R2019b pseudo-code for calculating the reliability polynomial coefficients N_{n-3} of the all-terminal networks.

Algorithm A1: The calculation of reliable polynomial coefficients for all-terminal networks

Input: The adjacency matrix A of a network with n vertices and m edges
Output: The value of coefficient N_{n-3} in the reliable polynomial of the network

```
Initialize m=0;
for i = 1 : n do
    for j = i : n − 1 do
        for x = j : n − 2 do
            C = A;
            A(:, i) = [];
            A(:, j) = [];
            A(:, x) = [];
            Calculate the path matrix P corresponding to A;
            Determine whether all elements in P are non-zero;
            if all(all(P)) then
                m=m+1;
            end
            A = C;
        end
    end
end
disp(m);
```

References

1. Liu, S.; Cheng, K.H.; Liu, X.P. Network reliability with node failures. *Networks* **2000**, *35*, 109–117. [CrossRef]
2. Archer, K.; Graves, C.; Milan, D. Classes of uniformly most reliable graphs for all-terminal reliability. *Discret. Appl. Math.* **2019**, *267*, 12–29. [CrossRef]
3. Romero, P. Building uniformly most-reliable networks by iterative augmentation. In Proceedings of the 2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM), Alghero, Italy, 4–6 September 2017; pp. 1–7.
4. Chen, Y.N.; He, Z.S. Bounds on the reliability of distributed systems with unreliable nodes and links. *IEEE Trans. Reliab.* **2004**, *53*, 205–215. [CrossRef]
5. Dong, L.; Zhao, H.; Lai, H.J. Local Optimality of Mixed Reliability for Several Classes of Networks with Fixed Sizes. *Axioms* **2022**, *11*, 91. [CrossRef]
6. Brown, J.; DeGagné, C.D.C. Roots of two-terminal reliability polynomials. *Networks* **2021**, *78*, 153–163. [CrossRef]
7. Bertrand, H.; Goff, O.; Graves, C.; Sun, M. On uniformly most reliable two-terminal graphs. *Networks* **2018**, *72*, 200–216. [CrossRef]
8. Shannon, C. Reliable circuits using less reliable relays. *J. Frankl. Inst.* **1956**, *262*, 191–208.
9. Hebert, P. Sixty years of network reliability. *Math. Comput. Sci.* **2018**, *12*, 275–293.
10. Politof, T.; Satyanarayana, A. A linear-time algorithm to compute the reliability of planar cube-free networks. *IEEE Trans. Reliab.* **2002**, *39*, 557–563. [CrossRef]
11. Provan, J. The complexity of reliability computations in planar and acyclic graphs. *SIAM J. Comput.* **1986**, *15*, 694–702. [CrossRef]

12. Provan, J.; Ball, M. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.* **1983**, *12*, 777–788. [CrossRef]
13. Politof, T.; Satyanarayana, A. Efficient algorithms for reliability analysis of planar networks—A survey. *IEEE Trans. Reliab.* **2007**, *35*, 252–259. [CrossRef]
14. Xiao, Y.; Zhao, H.; Mao, Y.; Chen, G. Invulnerability of planar two-tree networks. *Theor. Comput. Sci.* **2019**, *767*, 16–25. [CrossRef]
15. Bondy, J.A.; Murty, U.S.R. *Graph Theory*; Springer: New York, NY, USA, 2008.
16. Husain, A.; Nanda, M.N.; Chowdary, M.S.; Sajid, M. Fractals: An Eclectic Survey, Part-I. *Fractal Fract.* **2022**, *6*, 89. [CrossRef]
17. Husain, A.; Nanda, M.N.; Chowdary, M.S.; Sajid, M. Fractals: An Eclectic Survey, Part II. *Fractal Fract.* **2022**, *6*, 379. [CrossRef]
18. Zhang, Z.; Liu, H.; Wu, B. Spanning trees in a fractal scale-free lattice. *Phys. Rev. E* **2011**, *83 Pt 2*, 016116. [CrossRef]
19. Chang, S.C.; Chen, L.C.; Yang, W.S. Spanning Trees on the Sierpinski Gasket. *J. Stat. Phys.* **2007**, *126*, 649–667. [CrossRef]
20. Zhang, Z.; Liu, H.; Wu, B. Enumeration of spanning trees in a pseudofractal scale-free web. *Europhys. Lett.* **2010**, *90*, 68002. [CrossRef]
21. Skums, P.; Bunimovich, L. Graph fractal dimension and structure of fractal networks. *J. Complex Netw.* **2020**, *8*, 037. [CrossRef]
22. Gong, H.; Li, S. The number of spanning trees of a family of 2-separable weighted graphs. *Discret. Appl. Math.* **2017**, *229*, 154–160. [CrossRef]
23. Woodall, D. Tutte polynomial expansions for 2-separable graphs. *Discret. Math.* **2002**, *247*, 201–213. [CrossRef]
24. Sun, D.; Li, L.; Liu, K.; Wang, H.; Yang, Y. Enumeration of subtrees of planar two-tree networks. *Appl. Math. Comput.* **2022**, *434*, 127404. [CrossRef]
25. Joiţa, D.M.; Jäntschi, L. Extending the Characteristic Polynomial for Characterization of C20 Fullerene Congeners. *Mathematics* **2017**, *5*, 84. [CrossRef]
26. Li, T.; Yan, W. Enumeration of spanning trees of 2-separable networks. *Phys. A Stat. Mech. Its Appl.* **2019**, *536*, 120877. [CrossRef]
27. Yan, W.; Yeh, Y. On the number of matchings of graphs formed by a graph operation. *Sci. China Ser. A Math.* **2006**, *49*, 1383–1391. [CrossRef]
28. Zhou, J.; Bu, C.J. The enumeration of spanning tree of weighted graphs. *J. Algebr. Comb.* **2021**, *54*, 75–108. [CrossRef]
29. Xiao, Y.; Zhao, H. New method for counting the number of spanning trees in a two-tree network. *Phys. A Stat. Mech. Its Appl.* **2013**, *392*, 4576–4583. [CrossRef]
30. Yang, Y.; Klein, D. Resistance distance-based graph invariants of subdivisions and triangulations of graphs. *Discret. Appl. Math.* **2014**, *181*, 260–274. [CrossRef]
31. Yan, W.; Yeh, Y. On the matching polynomial of subdivision graphs. *Discret. Appl. Math.* **2009**, *157*, 195–200. [CrossRef]
32. Jin, X.; Zhang, F.; Dong, F.; Tay, E. Zeros of the jones polynomial are dense in the complex plane. *Electron. J. Comb.* **2010**, *17*, 2493–2503. [CrossRef]
33. Zhang, Z.; Zhou, S. Self-similarity, small-world, scale-free scaling, disassortativity, and robustness in hierarchical lattices. *Eur. Phys. J. B* **2007**, *56*, 259–271. [CrossRef]
34. Rozenfeld, H.; Ben-Avraham, D. Percolation in hierarchical scale-free nets. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2007**, *75*, 061102. [CrossRef] [PubMed]
35. Lin, Y.; Wu, B.; Zhang, Z.; Chen, G. Counting spanning trees in self-similar networks by evaluating determinants. *J. Math. Phys.* **2011**, *52*, 113303. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.