The Journal of Supercomputing (2022) 78:9741–9781
https://doi.org/10.1007/s11227-021-04216-2

Efficient methods for computing the reliability polynomials of graphs and complex networks

F. Safaei[1] · **R. Akbar**[1] · **M. Moudi**[2]

Accepted: 16 November 2021 / Published online: 19 January 2022

Abstract

Various methods have been proposed to evaluate the reliability of a graph, one of the most well known of which is the reliability polynomial, $R(G, p)$. It is assumed that $G(V, E)$ is a simple and unweighted connected graph whose nodes are perfect and edges are operational with an independent probability p. Thus, the edge reliability polynomial is a function of p of the number of network edges. There are various methods for calculating the coefficients of reliability polynomial, all of which are related to their recursive nature, which has led to an increase in their computational complexity. Therefore, if the difference between the number of links and nodes in the network exceeds a certain amount, the exact calculation of the coefficients $R(G, p)$ is practically in the NP-hard complexity class. In this paper, while examining the problems in the previous methods, four new approaches for estimating the coefficients of reliability polynomial are presented. In the first approach, using an iterative method, the coefficients are estimated. This method, on average, has the same accuracy as common methods in the related studies. In addition, the second method as an intelligent scheme for integrating the values of coefficients has been proposed. The values of coefficients for smaller, larger, and finally intermediate indices have been determined with the help of this intelligent approach. Further, as a third proposed method, Benford's law is utilized to combine the coefficients. Finally, in the fourth approach, using the Legendre interpolation method, the coefficients are effectively estimated with an appropriate accuracy. To compare these approaches fairly and accurately with each other, they have been carried out on synthetic and real-world underlying graphs. Then, their efficiency and accuracy have been evaluated, compared, and analyzed according to the experimental results.

Keywords Network reliability · Reliability polynomial · Robustness measure · Graph theory · Randomized algorithm · Numerical analysis

✉ F. Safaei
f_safaei@sbu.ac.ir

Extended author information available on the last page of the article

1 Introduction

In recent years, communication systems and infrastructures are mostly grappling with so many problems, while the speed and volume of communication and information have grown incredibly fast; the classical methods are incapable of solving them. These reasons have led to a great deal of attention and movement toward interdisciplinary trends such as complex networks based on graph structures. Such networks consist of a set of agents and a complex set of relationships between these agents. These structures can be considered as large graphs of nodes connected by one or more specific types of interdependencies. Most of these structures are very complex, and these networks are inherently so dependent on various fields of science that they cannot be analyzed with a view to a scientific discipline [1–4]. To develop any successful communication network, different aspects must be considered. Accordingly, important factors such as cost, security, integrity, scalability and fault tolerance can be mentioned. The latter factor is vital, especially for any communication network; hence, the issue of network robustness and resilience has gained considerable importance in the field of complex network research.

A system is robust when it has the ability to maintain its core functions even in the presence of external or internal errors. In the literature, robustness refers to the ability of the system to perform the main task, which, even after removing fraction of the nodes or edges from it, remains connected as far as possible from a global perspective. Further, resilience is defined as the ability of any entity in a network to stand against drastic changes in that network and its applications.

1.1 Motivations

Robustness in complex networks and modern infrastructures is one of the vital and fundamental features and has become one of the favorite and growing research fields in recent years. This field seeks to find the solutions and mechanisms to improve the connectivity of networks against random failures and systematic attacks. Therefore, various criteria to measure and evaluate the degree of resilience have been proposed in graphs and complex networks. The performance of a network can be greatly affected by its structural features. One of the main structural features is the robustness, which is expressed by terms such as resilience, fault tolerance, and survivability. In ecology, robustness is an important attribute of an ecosystem and can be used to provide insights into the response to disturbances such as the extinction of species and generations. In biology, the utility of network robustness is to study diseases and mutations in genes. In economics, resistance economics can increase our understanding of inflation, the risks of banking and marketing systems, and the like. Failures in the operation of transportation systems may have serious consequences and disruption to transportation. In the engineering science, robustness can also be useful for assessing the resilience of infrastructure networks such as Internet and power grid. Hence, the main motivation of many studies devoted to the design of robust networks arises from their vital applications in communication networks, power grids, transportation systems, sensor networks, logistics networks, etc. [5].

The robustness of any network is usually based on fault tolerance and vulnerability to random failures and targeted attacks. A large number of researches have been devoted to design of networks with optimal robustness.

Due to the lack of a comprehensive approach to the issue of resilience from a practical view, we have seen inappropriate and sometimes difficult and impractical solutions. In summary, the reasons for this attention as well as the large number of studies in the related work on the importance of network robustness can be summarized in the following two main objectives.

- Designing new networks of suitable and integrated systems, considering that in such networks failures may occur as well as there is a possibility of attacks by malicious individuals
- Protecting existing networks with the aim of identifying important and sensitive nodes and reducing their sensitivity by improving their resilience against random failures and targeted attacks

1.2 The problem statement

One of the obvious weaknesses in the research for robustness is the lack of a systematic approach. Any non-systematic approach to understanding the purpose of robustness and its challenges will lead to an ineffective view of robustness and, consequently, an inefficient solution.

Defining the purpose of resilience usually requires appropriate criteria. Although the topology of networks is an important criterion for understanding and characterizing complex networks, it alone is not sufficient to explain and capture the interactions and properties of networks, which are very wide and diverse. For this reason, a number of statistical criteria have been developed, each of which is relevant, applicable, and useful in different and specific fields. Thus, this field primarily seeks to find solutions and mechanisms to improve the resilience of complex networks against random failures and systematic attacks.

So far, various criteria have been presented to evaluate the robustness and resilience of graphs and complex networks. An important part of such criteria has been dedicated to the graph theory. In this manuscript, the reliability polynomial criterion has been utilized to scrutinize the robustness of graphs and complex networks. Additionally, a set of conceptual and practical tools to facilitate the analysis and evaluation of the robustness of graphs has been utilized by means of adapting them to the research contexts. Moore and Shannon first proposed the concept of reliability polynomial [6]. They introduced a probability model in which the network nodes are assumed perfectly reliable and links (edges) can be defective with a certain probability such as p. The main issue is to determine the probability that the network will remain connected under these conditions. If all links are operational with equal probability p, the reliability of the whole network is described in terms of a function of p, which in turn will lead to the polynomials of network reliability.

Unlike other robustness criteria that attribute a numerical value (usually normalized) to the robustness of a graph, the reliability polynomial advantage is that

a function in terms of p assigns to each graph. It is possible to use the robustness diagram of a graph and compare it with the other networks with the help of the plot of this function and determining the area under its curve. Hence, the result of the reliability polynomials of different graphs can be directly compared with each other. However, an important and fundamental challenge that has made the use of reliability polynomials directly difficult and even impossible is the calculation of their coefficients. It should be noted that the exact calculation of the coefficients is only possible for low-order graphs, and its exact solution falls into the category of NP-hard problems. In this paper, our main contribution is the application of various numerical and approximate techniques to estimate the reliability polynomial coefficients.

This article is organized into five sections. In Sect. 2, we briefly review the related work and model assumptions. Also in this section, the reliability polynomials are defined and we briefly study the methods of calculation the coefficients. In Sect. 3, we present the proposed methods with details. The numerical results of the proposed methods on several synthetic and real-world graphs are described in Sect. 4. In Sect. 5, the validation of various measures is examined. Moreover, the performance of the proposed approaches is simulated and evaluated in this section. In Sect. 6, robustness analysis is presented to assess the resilience of graphs and networks, along with insights into classifying the classes of robustness metrics. Some of the most important measures belonging to each category are defined and then for the graphs studied in this research; numerical results are presented along with the analytical report. Finally, the conclusions are summarized in Sect. 7, and we suggest future work to continue the current study.

2 The literature

Moore and Shannon [6] first proposed the reliability polynomials. The aim of these researchers was to study to improve the reliability of replacing an electromechanical relay with two continents (switching device) through a network consisting of n similar relays by creating a connection between input and output. If the coil was not energized, there would be a path with a probability greater than the assumed value of c, and conversely, if the coil was energized, with a probability less than the assumed value of a ($a<c$), there would be a path between input and output. They introduced a probability model in which network nodes were assumed perfectly reliable, but the links (edges) could fail with a hypothetical probability such as p.

Using this concept, Cowell et al. [7] have carefully examined the reliability of small hammock networks. They showed that array-based designs such as vertical FET (VFET), vertical slit FET (VeSFET), FinFETs, and nano-electromechanical systems (NEMS), as well as arrays of CMOS devices, could fit well into small hammock networks. Rohatinovici, Proştean, and Balas [8] showed that communications in an axon can be modeled using an array of logical gates so that each logical gate emulates a voltage-gated ion channel. They estimated the probability of correct communication in such a model by analyzing the reliability for logical gate circuits as probabilistic gate matrix (PGM). Thus, another reliability polynomial application

can be used to estimate the reliability of molecular networks for processing information inspired by biological systems.

Robledo et al. [9] have developed an interpolation-based technique for estimating reliability polynomial coefficients. Their idea is to use the interpolation with Newton's method and then to correct its coefficients using the Hilbert $L^2[0, 1]$ space. To ensure the correctness of the proposed method, they used the equality of the number of spanning trees with the polynomial coefficients corresponding to the smallest polynomial power. Then, they present that, at best, the complexity of the calculation can be done in polynomial time and that the reliability polynomial has zero root from the order of repetition n-1 (n is the order of the graph). In addition, Burgos and Amoza [10] have proved a new general theory of reliability polynomials.

Using reliability polynomial, Khorramzadeh et al. [11] have analyzed the effect of network structural motifs on diffusive dynamics such as the spread of infectious disease. In addition, Eubank et al. [12] have used the reliability polynomial to characterize and design networks and have defined a new criterion based on node and edge centrality called criticality and shown its relationship to the binary networks. They point out that the application of network reliability polynomial can be used to target interventions to control outbreak.

Brown and two co-authors [13] have introduced a measure called the average graph reliability, which is the ATR (All-Terminal-Reliability) integral over the range [0, 1]. Their proposed benchmark is an alternative to uniformly reliable network. In fact, due to the accurate calculation of the reliability polynomial coefficients, that is a NP-hard problem, the authors have proposed strategies for limiting the average reliability that do not necessarily require the calculation of the reliability polynomial.

Brown and Dilcher [14], as well as Brown and Colbourn [15], describe about the location of the reliability polynomial roots. Brown and Colbourn [15] showed that all true roots of a reliability polynomial are located on a complex plane and a single disk centered on 1 ($|z\text{-}1| = 1$ or exactly in the range $0 \cup (1, 2]$). They also conjectured that all the complex roots were in z: $|z\text{-}1| \leq 1$, which Royle and Sokal [16] rejected.

Brown, in collaboration with Koç and kooij [17], has determined the inflection points to measure the reliability of networks with the help of the reliability polynomial. They show that there is at most one inflection point in the range (0, 1) and there are families of simple graphs that may have more than one inflection point. The same authors [18] examine families of graphs whose the reliability function may cross more than twice with the x-axis. Page and Perry [19] have provided a way with the network reliability polynomials to rank the important and critical edges in networks. Chen et al. [20] also wrote a short note on this article and pointed out the problem of redundancy in defining edge rankings based on the number of spanning trees.

Beichl and Cloteaux [21] have proposed a randomize approximation algorithm for calculating the reliability polynomial coefficients. They have shown that their proposed method can have an empirically faster convergence rate compared to the Colbourn's approximation method [22]. Beichl, in collaboration with Harris and Sullivan [23], proposed a SIS-based (Sequential-Importance-Sampling) bottom-up algorithm for approximate estimation of the reliability coefficients by selecting

a spanning tree and adding an edge. The algorithmic complexity of their research is around $O(|E|^2)$, and the SIS technique is utilized to reduce the variance. In another research, Harris, Sullivan, and Beichl [24] provide a method for accelerating and improving a proposed SIS-based method, which is a general technique for estimating a particular distribution that provides examples of a distribution function rather than a given distribution. For graphs with m edges and n nodes, the complexity of their research is reported of the order $O(m \log n\, \alpha(\mathrm{m,n}))$.

From what we have summarized above, it is obvious that after half a century of the seminal research of Moore and Shannon [6], extensive research in this field has been reported and accumulated. In addition, many variants of this original problem have been examined, generalized, and analyzed. We properly review the concepts, research, and important results related to the present article, as well as some of the most important developments in this field. For this reason, we have mentioned only some of the most important research directions. It should be noted, however, that several methods have been proposed in the literature for calculating the reliability polynomial coefficients, and it could not be overlooked that these calculations will become intractable as the size and order of the graph increase.

2.1 The model assumptions

The structure function of the coherent network system is defined as a function of its components, and it is assumed that the network has $m \geq 1$ edges, which each edge can be operational with an independent probability p. To describe this situation, we define a binary variable x_i.

Definition 1 [25]: *The binary function* $\varphi(\vec{x}) : \{0,1\}^m \rightarrow \{0,1\}$ *is defined as the network structure function and* $\vec{x} = (x_1, \ldots, x_m)$ *is called the binary vector of its links' state, whenever.*

$$x_i = \begin{cases} 1 \text{ edge } i\text{th operates} \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

$$\varphi(\vec{x}) = \begin{cases} 1 \text{ network operates} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

In fact, the structure function of the network represents the relation among network and its components. Once the connection between the edges has been identified in the network, the form of the function $\varphi(\vec{x})$ will be determinable. A network with m edge-component has 2^n possible states for the edge vector. Depending on the network type, the status vector indicates that the value of the system structural function is 0 or 1. Although the network structure function can have no behavioral limitations mathematically, it is limited to the monotone functions, caused by expressing the behavior of the network.

Definition 2 [25]: *A network with the structure function is monotone where for two state vectors of x and y (for all edges $x_i \leq y_i$), $\varphi(x) \leq \varphi(y)$.*

The above definition means that in a monotone network, the reliability of the network will not be worse when one of edges becomes operational. Of course, we will not consider such an assumption in this paper because considered networks are not repairable.

Definition 3 [25]: *Edge i in the network is an irrelevant edge when its presence/absence has no effect on φ. In other words.*

$$\varphi(x_1, ..., x_{i-1}, 0, x_{i+1}, ..., x_m) = \varphi(x_1, ..., x_{i-1}, 1, x_{i+1}, ..., x_m)$$

Definition 4 [25]: *A network is coherent if and only if its structure function is monotone and there is not an irrelevant component in it.*

It is assumed that the state of the edge i in the graph is a binary random variable with the following probability mass function.

$$P\{X_i = x_i\} = \begin{cases} p_i & \text{if } x_i = 1 \\ 1 - p_i & \text{if } x_i = 0 \end{cases} \tag{3}$$

where p_i is the probability that edge i is operational (reliable), and we get

$$p_i = P\{X_i = 1\} = E[X_i]$$

There are three issues with network reliability, two-terminal reliability or 2TR, all-terminal reliability or ATR and k-terminal reliability or KTR [6]. In 2TR, it is assumed that there exist two distinct nodes s (source) and t (terminal or target), and then, given the probability of p (operating one edge), we calculate the probability of existing an operational path across s and t. In case of ATR, the problem is to find an operational path between the two desired nodes u and v of the network. Finally, KTR points out that if we have a set of K with distinct nodes $|K| = k$, what is the probability of an operational path between the two nodes $u, v \in K$?

All three of the above are pertain to domain analysis, and they are NP-complete. It should be noted that the NP class contains problems, which can be solved by a Turing machine with non-deterministic counting in polynomials. If it is NP-complete, therefore it is NP-hard. In this paper, the goal of the proposed methods is to calculate the reliability of the entire network terminal, which is called ATR, as defined below.

$$P\{\varphi(x) = 1\} = E[\varphi(x)]$$

In fact, the probability of being connectedness for a graph is assumed to be perfect nodes and the probability of links' failure. Since φ is a function of the state vector of network G, the reliability of G is a function of the vector $P = (p_1, \ldots, p_n)$ represented by $R(G, P)$. In this paper, it is assumed that the probability of being

operational for all edges is the same, independent and equal to p, therefore the same symbol $R(G, p)$ can be used to denote the polynomials reliability of the graph G.

It is assumed that we are facing a stochastic graph with the probability of links failure. At a given time, each edge can be in one of two possible states: Up (operational) or Down (failed), which is called binary networks [25]. The probability of being in the first state is p and the probability of being in the second state is $1–p$.

Moreover, it is assumed that p remains constant over all time for all edges $e \in E$ and also, for all $f, e \in E, f \neq e$, the failure probability of each edge such as f is independent of other edge such as e.

In addition, after each failure at an edge, there is not any reconnection among its related nodes. To design a simpler model, we assume that the probability of failure is the same for all edges of the network. Also, we assume that the capacity of all links is infinite or alternatively. It means that the information traveling through those links on the network is negligible compared to their capacity. The rationale behind this assumption is to preclude the possibility of occurrence cascading failures may make our reliability analysis intractable.

2.2 Reliability polynomials

Definition 5 [6]: *Assuming a criterion for network functionality, each pathset is a subset $O \subseteq E$ of edges that makes the graph connected; in other words, G (V, O) is operational. Therefore, if our problem is ATR, as mentioned above, the pathset will be any connected spanning subgraph of G. A minimal pathset (based on the inclusion principle) is called minpath. In 2TR, the minpath is only one path between the source and destination nodes, but for ATR, the minpath is a spanning tree, and in KTR, the minpath is a Steiner tree.*

Definition 6 [25]: *The set of all indices of the state vector x where $x_i = 0$ is called cutset, $C = \{i: x_i = 0\}$.*

Thus, the cutset is a subset $C \subseteq E$ of edges that the subgraph $G'(V, E \backslash C)$ is not connected. If the two vectors x and y are the network state vectors with $x < y$ and $\varphi(y) = 1$, then x is called the mincut vector. Further, the mincut vector is the vector for which the network is disconnected and, if at least one of its edges is operational, the network may be re-operational.

Definition 7 [25]: *If $\varphi(x)$ is the structure function of a coherent network, the state vector x is the path vector and the set of indices where $x_i = 1$ is the pathset whenever $\varphi(x) = 1$. In other words, if $P = P(x)$ is the set of paths, then $P = \{i: x_i = 1\}$.*

Definition 8 [25]: *Let x, y be two vectors for which $y < x$, $\varphi(y) = 0$; then, vector x is called the minimal path vector and the corresponding pathset is called the minimal pathset.*

Hence, the minimal path state vector is the vector for which the network is connected and inactivation of at least one of its operational links will cause the whole network to be disconnected.

Definition 9 [6]: *Let m be the number of edges (size) of G (V, E). Suppose* $G'(V, E')$ *for* $E' \subseteq E$. *The reliability polynomials are defined as follows.*

$$R(G,p) = \sum_{\substack{E' \subseteq E \\ G' \text{ is a pathset}}} p^{|E'|}(1-p)^{m-|E'|} \tag{4}$$

Consequently, $R(G, p)$ denotes the probability of the graph G which is a function of p, and it is truly written based on a polynomial of p. Note that the definition of reliability polynomials is the same for 2TR, KTR and ATR, and the only difference is in the definition of pathsets and cutsets.

There are various forms to define the reliability polynomials.

Definition 10 (N-form) [6]: *Let* N_i *be the number of pathsets containing i edges. The probability of making a set of i edges that causes connectivity is.*

$$R(G,p) = \sum_{i=0}^{m} N_i p^i (1-p)^{m-i} \tag{5}$$

Definition 11 (C-form) [6]: *If the number of cutsets with i edges is assumed to be* C_i, *we have.*

$$R(G,p) = 1 - \sum_{i=0}^{m} C_i p^{m-i}(1-p)^i, N_i + C_i = \binom{m}{i} \tag{6}$$

Definition 12 (F-form) [6]: *An F-set is a set of links that a pathset is its complements. F-sets can recommend another approach to define the reliability polynomials.*

$$R(G,p) = \sum_{i=0}^{m-n+1} F_i p^{m-i}(1-p)^i, F_i = N_{m-i} \tag{7}$$

where F_i denotes the number of subdomains or subgraphs that have exactly i edges these are remained $m-i$ edges after deletion and their deletion causes G to be disconnected. The vector $\langle F_0, F_1, ..., F_{m-n+1} \rangle$ is the F-vector of graphic matroid of G, i.e., a simplicial complex of subsets $E' \subseteq E$ that their deletion causes G disconnected [26]. It should be noted that we apply F-Form in this paper.

The reliability polynomials can be used to compare the reliability of different graphs. However, the important point to note is that the reliability polynomials do not represent a total order in the set of graphs and define a partial order.

Although the ATR criterion seems simple at first glance, there consist many problems. Unlike other robustness measures that assign a numerical value (usually normalized) to the graph's robustness value, the reliability polynomials assign a function in p to each graph. Accordingly, one of the major problems in the ATR is the allocation of the appropriate value to p, and the evaluation of the graph's robustness depends on the p value, so it may be difficult or even impossible to compare

the reliability measure of the two different graphs directly. For example, it may be difficult to compare the reliability polynomials of which the robustness value of two graphs G and H is very close in vertices and edges.

In such cases, the plot of the difference between their reliability polynomials can compare better. For example for $p \to 0$, $R(G, p) > R(H, p)$ (or vice versa) while for $p \to 1$, $R(H, p) > R(G, p)$ (or vice versa) [13].

In ATR applications, with assuming a fixed number of nodes and edges in order to compare different networks and select which is most robust, the first case, $p \to 0$, the probability of the links failure is high, and accordingly, ATR acts as a criterion for the number of spanning trees. A graph can be chosen with maximum spanning trees (such a graph near $p = 0$ is optimal); however, this assumption is not true because this probability is not too high in most networks.

According to [6], Shannon and Moore point out that the reliability polynomials for $p \to 1$ always provide a similar assessment to the edge connectivity, although it may omit the effect of adding edges to the network. In other words, a graph with the largest edge connectivity and the least number of cutsets with the minimum size (such a graph would be optimal near $p = 1$) will be chosen. However, the ATR criterion gives a more realistic interpretation of the robustness when $p \to 1$. Hence, it is recommended to use the reliability polynomials at different intervals, especially $p \to 1$ for evaluating the network robustness; because in real-world networks, there is rarely links' failure [27].

On the other hand, when there is no knowledge of the value of p, how can one make a decision? In [13], Brown and his colleagues have introduced and evaluated the average reliability measure of the graph, which is ATR integral in the range [0, 1]. Their proposed criterion is, in fact, an alternative for uniformly reliable networks. Although the average reliability computation is NP-hard and its calculation is as difficult as the reliability polynomials computations, these authors have provided some strategies to moderate the average reliability that do not necessarily require the reliability polynomials computation.

2.2.1 Calculating the coefficients of reliability polynomial

In this section, we briefly review some of the computation methods of reliability polynomials' coefficients. Then, we describe in more detail the proposed methods, which attempt to accurately compute each polynomial coefficient.

If l is the minimum cardinality pathset, then for $i < l$, $N_i = F_{m-i} = 0$. In 2TR, a pathset with minimum cardinality is a short path between s and t that is computed in linear time. In ATR, this is equivalent to a spanning tree, so $l = n - 1$. Finally, in KTR, the minimal Steiner tree will be specific for k vertices, which is NP-complete. Thus, in 2TR, N_l is the number of short paths and it is equal to the number of spanning trees (tree number or graph complexity) in ATR; it is a solution to calculate the number of these trees using Kirchhoff's matrix theorem [28].

$$N_l = (-1)^{i+j} \det(L_{i,j}) \tag{8}$$

In fact, $N_l = \kappa(G)$ (the edge connectivity of the graph G) in which $\kappa(G)$ is the minor of Laplacian matrix, defined as $L = D - A$, where D is the degree diagonal matrix and A is the adjacent matrix of graph G. This computation can be done in the polynomial time.

Clearly, the reliability polynomials are a polynomial with integer coefficients. Satyanarayana and Chang [29], using the dominating set theory, proved that the coefficients' signs of these polynomials in the intermediate variable p will change. If c is a minimum cardinality cutset, then for $i < c$, $C_i = 0$. Similarly, $N_{m-i} = \binom{m}{m-i}, i < c$. Using the network flow algorithm that is repeated for each pair of vertices, the parameter c can be computed in the polynomial time. Thus, we get

$$\underbrace{N_0, ..., N_{l-1}}_{N_i=0}, \overbrace{N_l, N_{l+1}, ..., N_{m-c-1}}^{\text{unknown}}, N_{m-c}, \underbrace{N_{m-c+1}, ..., N_m}_{N_i=\binom{m}{m-i}} \tag{9}$$

Other coefficients may be computationally efficient, but the complexity of the 2TR, ATR, and KTR is different. As we know, the F-set is a set of links that complement a pathset and *Fi* is the number of these F-sets with the size of i edge. Because every set of links is either a cutset or an F-set, so $F_i + C_i = \binom{m}{i}$ and if we consider $d = m - 1$, there is

$$\underbrace{F_0, ..., F_{c-1}}_{F_i=\binom{m}{i}}, F_c, \overbrace{F_{c+1}, ..., F_{d-1}}^{\text{unknown}}, \underbrace{F_{d+1}, ..., F_m}_{F_i=0} \tag{10}$$

It should be noted that in the ATR, which we consider in this paper, the values of l, c, N_i and N_l are computable at the polynomial time. In addition, the parameters N_{m-c-k}, N_{m-c} and N_{m-i} can be computed at the polynomial times. However, the calculation $\sum_{i=0}^{m} N_i$ of the order of complexity is NP-complete [5].

If we can identify minpaths, then we can compute the reliable polynomials using the principle of inclusion–exclusion. Assuming that P_1, P_2, …, P_h are pathsets, the probability that all edges of the minpath P_i are operational is presented with E_i; then, probabilities of occurrence of one or more than one E_i are:

$$R(G, p) = \sum_{j=1}^{h} (-1)^{j+1} \sum_{\substack{I \subseteq \{1,2,...,h\} \\ |I|=j}} \mathrm{P}\{E_I\} \tag{11}$$

3 The proposed methods

Up to now, various methods have been proposed to calculate the coefficients of reliability polynomials discussed in Sect. 2. The problem with all of these algorithms lies on their recursively, which increases their computational complexity. If the

number of links (or more precisely the difference between the number of links and nodes) exceeds a certain level, the accurate calculation of the reliability polynomials becomes practically an integral problem. Thus, it can be concluded that almost none of the algorithms for calculating the reliability polynomials coefficients can be useful for a complex network, especially with large dimensions.

In general, we have methods based on stochastic algorithms (approximation) as well as the use of interpolation for inaccurate calculation of the reliability polynomials. In the following, we present new ideas for both approximate and interpolation-based methods, and then, we evaluate these ideas through next sections.

3.1 The approximation approaches

As mentioned before, the reliability polynomials can be overwritten in the various forms, which we have shown in Sect. 2 (Definitions 10–12). We also note that these forms can be converted to each other, and because of the continuity of the F-form method (Eq. (7)), we use it as an equation for computing the exact coefficients of reliability polynomials. In Eq. (7), m is the number of edges and F_i is the number of connected subgraphs obtained by removing i edge from the graph.

For a given graph, which has m edges and n nodes, it can be seen that if we remove more than $m-n+1$ edges from the graph, then the number of edges is smaller than $n-1$ and the graph is completely disconnected. This means that in all graphs, F_i will be zero for i greater than $m-n+1$. In general, we need to calculate the F_i recursively. To calculate F_n in a loop, remove an edge at each time, calculate F_{n-1} for the resulting graph and then add all together.

The estimated algorithms can be divided into two general categories: top-down and bottom-up [21, 23]. All of these algorithms estimate F_i and then substitute F_i in F-form equation (Eq. (7)) to obtain the final polynomial.

The top-down algorithms start from the graph itself and remove one edge from the graph each time to reach a tree. However, in bottom-up algorithms, it starts from a spanning tree and every time an edge is added to the tree. This is precisely why top-down algorithms calculate the low coefficients of the reliability polynomials (or F_i) with small i. Similarly, the higher coefficients (F_i with large i) in the bottom-up algorithms have more accurate computation.

In the top-down algorithms [23], m–$n+1$ edges are eliminated by a uniform probability distribution and randomly from the graph, at first. Then, at each step, the number of edges that can be deleted and the graph remains connected is stored. Finally, by multiplying these values together and dividing the product by the permutations, we will be able to estimate F_i. Simply, instead of the computing to remove each edge, one edge can be represented as the rest of the edges, and instead of the tree being fully produced, only one path from root to leaf produced. Since this is ultimately done many times, their average will be calculated. It is expected that the estimated F_i is closer to actual F_i. However, as the index i becomes larger, the difference between the estimated value and the actual value increases.

The general approach of the bottom-up algorithm is similar to the approach of the top-down algorithm. However, the difference is that a spanning tree is selected from

the graph at first, and then, one edge is added each time in the bottom-up algorithms. Augmenting an edge is based on a probability, and unlike the top-down algorithm, it does not happen randomly. According to starting of a spanning tree and the estimation operation, the larger the index i, the smaller the difference between the estimated value and the actual value [21, 23].

In summary, in both top-down and bottom-up approaches, if the difference between the number of edges and nodes in a given network is very high, the estimated polynomial difference with the original polynomial is very large, too. Because it will increase the primary, (secondary) F_i will be very different from their actual values. In this paper, by proposing efficient methods, our main goal is to reduce this fundamental problem in estimation methods.

3.2 Methods based on random algorithms

In the previous section, we analyzed the problems with top-down and bottom-up approaches. In this section, we aim to illustrate the approximating and random algorithm-based methods for solving these problems. At the beginning, we express a new approach to compute F_i. Then, we present some efficient approximating methods how we combine F_i from different methods (top-down, bottom-up, and the proposed method) to achieve Fi that have the least difference with their actual values. These methods are discussed and expressed in this section. We analyze them later in the networks with the sufficient size to calculate the values of F_i and the reliability polynomials, accurately.

3.2.1 Estimating the F_i values: a novel iterative method

In the original F-form (see Eq. (7)), F_i values denote the number of connected subgraphs that emerged by removing i edges from the graph. Using this definition, we try to estimate F_i accurately. Assuming that m edges exist in a given graph; then, the total number of subgraphs with the elimination of i edges (with $m-i$ edges) will be equal to $\begin{pmatrix} m \\ m-i \end{pmatrix}$.

To identify F_i, one of the solutions is the drawing all of these subgraphs and then the enumeration of their connectivity number. However, such a scheme would occur extremely time-complexity due to the dramatic increasing the connectivity number.

Thus, another way to tackle this problem is to select i edges of the network randomly with assuming to have the *iteration_no* and then remove them. Next, we enumerate which subgraphs that equals to *iteration_no* (may also be iterative) have the connectivity feature, we display them as *connected_no*. The quantity of F_i is calculated by replacing the above values in Eq. (7). Since the values of F_i are necessarily integers, then it is essential to round them. Therefore, we get

$$Fi = \lceil \frac{\text{Connected\_no}}{\text{Iteration\_no}} \times \begin{pmatrix} m \\ m-i \end{pmatrix} \rfloor \tag{12}$$

where the meaning of symbol $\lceil \cdot \rfloor$ is the rounding operator.

The main challenge in this scheme is the number of iterations (actually the *iteration_no*) that needs to remove an edge from a given network. If this value is large, it may increase the complexity even more than the exact computation. Even if this value is negligible, it may lead to an incorrect result that is very different from the correct values. The idea behind this approach is derived from the fact that a number of samples from a statistical population are randomly selected and claimed to represent the whole population.

We considered this number as a constant number (for example, 100), at first. Then we observed during some experiments for a given graph with 10 edges, to remove one edge, the elimination process would be performed 100 times. However, if we chose all the modes, we only needed to do 10 times.

For this reason, an intuitive consequence is that the *iteration_no* depends on $\binom{m}{m-i}$. It should be noted that a maximum state is selected once. Hence, *iteration_no* can be calculated as:

$$\text{iteration\_no} = \sqrt{\binom{m}{m-i}} \tag{13}$$

The experimental results for the small-sized networks, which have accurate values of their reliability polynomial coefficients, are presented in the performance evaluation section (see Sect. 4). Further, a new method is compared with top-down and bottom-up approaches. The proposed scheme is referred as an *iterative method* because of its duplication.

3.2.2 Combining F_i values obtained from different approaches

In this section, we aim to come up with ideas for combining the F_i's captured from different approaches (top-down, bottom-up, and iterative method which is presented in the present study). The goal is to make the final F_i as close to the actual F_i as possible. To combine the desired values together, two different ideas are presented in this section.

3.2.2.1 Smart merging of F_i's Usually, when we have a set of values and want to use them to reach the final values, the simplest way can be the average. However, there is a set of information on F_i derived from each method that can be used more precisely to obtain the final F_i. The following information about F_i derived from different methods is.

- According to the argument in the current section, the value of F_i for small i in the top-down algorithm is closer to the actual values
- In the bottom-up approach, the F_i value for large i is closer to the actual value
- In the proposed iterative scheme in the present study, since the estimation is performed independently of the start of a tree or graph as well as the deletion/addition of edges, it can be argued that the differences obtained from F_i of the real

values follow a uniform distribution. Accordingly, it can be claimed that each part of F_i is approximately a distance equal to the actual value

Based on the above information, a method for integrating F_i's can be presented. Small F_i from the top-down algorithm, large F_i from the bottom-up algorithm, and middle F_i from the proposed iterative approach are extracted. The significant point in this scheme is to find the appropriate locations (i.e., the indices i that should be used) to perform the integration process.

Obviously, these indices depend on the number of F_i. Next, we designate these two values as low and high threshold. As mentioned earlier, in all graphs, the values of F_i for $i > m-n+1$ are zero. On the other hand, for all connected graphs, which the reliability polynomials are defined, $F_0 = 1$. For this reason, it is necessary for each graph to calculate F_i for $1 \leq i \leq m-n+1$ whose number is $m-n+1$.

Through simulation experiments, it can be seen that the best results are obtained if the following values assigned to the thresholds

$$\begin{cases} \text{Threshold}_{\text{low}} = \lceil 0.3 \times (m-n+1) \rfloor \\ \text{Threshold}_{\text{high}} = \lceil 0.7 \times (m-n+1) \rfloor \end{cases} \tag{14}$$

where $\text{Threshold}_{\text{low}}$ and $\text{Threshold}_{\text{high}}$ refer to the upper and lower thresholds, respectively; m is the number of edges and the parameter n is the number of nodes in the graph. Moreover, the meaning of symbol $\lceil \cdot \rfloor$ is the rounding operator.

The following equation is employed to integrate the F_i's and reach their final values.

$$F_i = \begin{cases} 1 & i = 0 \\ \text{Top\_Down} & 1 \leq i \leq \text{Threshold}_{\text{low}} \\ \text{Iterative} & \text{Threshold}_{\text{low}} < i < \text{Threshold}_{\text{high}} \\ \text{Bottom\_Up} & \text{Threshold}_{\text{high}} \leq i \leq m-n+1 \\ 0 & i > m-n+1 \end{cases} \tag{15}$$

Figure 1 depicts the pseudo-code for the integration approach of the three mentioned algorithms. The input is a given graph, and the output is a vector of F_i values. The parameters m and n represent the number of edges and nodes in the graph, respectively. As can be seen, only the required number (and not all values) of the F_i values is calculated for each algorithm. This will not cause an unnecessary increase in the execution time of the proposed integration process.

In the pseudo-code, Lines 9–18 of the integration process compute the top-down method for values of i smaller than $\text{Threshold}_{\text{low}}$. Then, the graph is returned to its original state (Line 19) and from lines 22 to 28, the applied algorithm is as the same as the bottom-up approach. From Lines 29 to 40, the calculation method is the proposed iterative scheme. Finally, the achieved vector, which is formed gradually during the execution of its code, is the result of the algorithm that be returned by the function.

It is common to execute the approximate algorithms to compute F_i up to 100 or 1000 times and then average the values obtained and, after rounding, the final F_i

```
1.  Vector CombinationAlgorithm (Graph)
2.  begin
3.     backup_graph = Graph
4.     Threshold_low  = Round(0.3 × (m − n + 1))
5.     Threshold_high = Round(0.7 × (m − n + 1))
6.     iteration _ no = sqrt( C(m, m − i) )
7.     result [0] = 1
8.     a_0 ← 1;
9.     for (k =1 to Threshold_low)
10.    begin
11.       vector temp = all edges which if removed graph do not disconnect Graph;
12.       a_k= size (temp);
13.       uniformly select an edge from temp and remove it from Graph;
14.    end
15.    for (k=1 to Threshold_low)
16.    begin
17.       result[k] = ∏_{0≤i≤k} a_i / k!
18.    end
19.    Graph = backup_Graph
20.    mm ← 1;
21.    kapa_g = number_spanning(Graph);  //Choose a spanning tree H of Graph uniformly;
22.    for(k= m-n+1 to Threshold_high)
23.    begin
24.       result[k] = mm × kapa _ g × C(m − n + 1, k)
25.       s = Σ_{d∈Graph−H} number _ spanning(H) / number _ spanning(H ∪ d)
26.       mm = mm × s / k
27.       Choose an edge from Graph-H with probability 1 / number _ spanning(H ∪ d) and add to Graph
28.    end
29.    for (k= Threshold_low+1 to Threshold_high-1)
30.    begin
31.       for(i=0 to iteration_no)
32.       begin
33.          Graph = backup_Graph
34.          connected_no = 0
35.          uniformly select k edge from Graph and remove them from Graph;
36.          if Graph is Connected
37.             connected_no++
38.       end
39.       result[k] = Round[ connected _ no / iteration _ no × C(m, m − i) ]
40.
41.    end
42.    return result;
   end
```

Fig. 1 Pseudo-code for integration approach

to calculate the polynomial is replaced in the main function [21, 23]. In the above idea, we have done the same thing and implemented the integration approach up to 1000 times. Obviously, the higher number of executions, the higher precision of the resulting polynomials. To make the comparison of the results fair, we implemented other schemes similar to this method as many times. In the following section, it is shown that the integration process will perform better and more accurate in comparison with a number of graphs.

3.2.2.2 Integrating the coefficients according to Benford's law In recent years, many articles about Benford's law have been published in a variety of fields including accounting, computing, statistics, etc. [30]. In the proposed approaches, usually the digit on the left of the number as a random variable follows the uniform discrete distribution, and the probability of each digit at this location is one-ninth. However, in some cases, the starting digit has a right-skewed discrete distribution in which the digit 1 has the highest probability; the least probability belongs to digit 9.

Related studies have shown the Benford's law as an example in electricity subscriber accounts, stock prices, population numbers, death statistics, the number of candidates in constituencies, and even polling stations, length or width of the river, financial bills and tax returns, Fibonacci series, molecular weight of materials, etc. [31]. A number of articles have been presented to detect fraud due to create numbers or information verification in different domains of this field.

If the random variable D_1 represents the first digit on the left of a given number, then according to Benford's law, D_1 has a discrete distribution as

$$P_{D1}(d_1) = \log_{10}(1 + \frac{1}{d_1}) \ , \ d_1 = 1, 2, ..., 9 \tag{16}$$

In 1995, according to Hill's research [32], it was found that this rule was not limited to the first significant digit and could be extended to higher digits. In order to apply the Benford's law in a set, it is always necessary to prove that this set can comply with the Benford's law or not. In this study, we consider a very simple case of Benford's law, the probability of the most valuable digit.

In some cases, the lack of enforcement of the Benford's law is quite intuitive. For example, if we consider the age of politicians in a society as a statistic sample, then the probability of 1 as valuable digit of their age (that means between 10 and 19 years or more than 100 years) is almost zero, and it is more likely to reach 4 or 5 numbers. In these circumstances, it is reasonable to observe that such a sequence of numbers does not follow Benford's law. Where this is not the case, it is usually possible to reject/prove the compliance by simulating, collecting, and analyzing a large amount of data.

For the problem addressed in this paper, which is achieving the final values of F_i computed in three methods, if it can be proved that the numerical values of the F_i or even the final coefficients, in terms of a set of numbers, follow Benford's law, an algorithm can be presented for their optimal integration.

To achieve this goal, we analyzed a large number of graphs that their reliability polynomial and their corresponding F_i coefficients were determined. It is

Table 1 Different values of the Benford's law

Digit	1	2	3	4	5	6	7	8	9
Benford's law (%)	30.1%	17.6%	12.5%	9.7%	7.9%	6.7%	5.8%	5.1%	4.6%

concluded that the F_i values do not comply with Banford's law because they are numerical and countable. However, it is interesting to note that the final reliability polynomial coefficients fully comply with this law.

It is difficult for us to prove that reliability polynomial coefficients are consistent with Benford's law. If the numeric values of F_i followed this law, it would be easy to choose a combination of F_i that is closest to Benford's values (it means less difference). However, in the current situation, it is required to identify the final polynomials, and finally, we ought to choose the case whose the final coefficients are more in line with Benford's law.

The number of F_i that equals $m-n+1$ can be determined from three different approaches (namely top-down, bottom-up, and the proposed iterative scheme). By combining these three sets of F_i, 3^{m-n+1} reliability polynomials can be obtained, which are probably distinct. This distinction is due to the probability of having the same value for the estimated F_i by two different methods for a given i. In this case, selecting each of them will lead us to the same polynomial.

Note that 3^{m-n+1} means three possible choices for each Fi that are multiplied together (the principle of multiplication). If $m-n+1$ is small, the corresponding reliability polynomial can be computed, and finally, the polynomial is selected whose coefficients have the greatest features of Benford's law. However, if $m-n+1$ is a large number (otherwise, it does not need for approximate computation), one has to think of another solution.

When $m-n+1$ is a large number, the final polynomial in all three methods is computed. Then, it is necessary to see which of the final coefficients of these polynomials most in line with Benford's law are. To do this, it is essential to count the number of coefficients starting with one, two, three, etc., and divide it by the total number to capture the percentages of each one. Then, the difference of these percentages with the Benford numbers (listed in Table 1) is calculated. Clearly, to determine the difference, the absolute value should always be considered.

After finding the best sentence out of the three existing sentences, we set it as the basis of the work. For this selection, we have three different choices.

1. If the sentence closest to Benford's values was the polynomials from the top-down approach, keep the first 10% of the F_i, change the last 10% of them with F_i of the bottom-up approach and consider these F_i as the basis. For the middle 80% of the F_i, we replace each with the other two F_i, calculate the polynomials, and continue with this polynomial if the difference is lower than Benford values. The number of check states can be reduced from 3^{m-n+1} to $2 \times 0.9 \times (m-n+1) = 1.8 \times (m-n+1)$. The initial values of 80% of the middle

are predicted to be appropriate; we begin this checking cycle separately for the 80% of the middle from the end of this range.

2. If the sentence that was closest to Benford values was due to the polynomial of the bottom-up scheme, then it would hold the last 10% of the F_i, and the first 10% would replace with F_i of top-down method and consider such F_i as the basis. For 80% middle of F_i, we do exactly like case 1. Note that this case is not the same as the previous one because the F_i considered, as basis in cases, 1 and 2 are different. On the other hand, for Case 2, we start checking for 80% of the middle from the beginning of same range.
3. If the sentence closest to Benford values is derived from the proposed iterative scheme, then we replace the first 10% of the F_i with the top-down method values and the last 10% of the F_i with the bottom-up values. Now, we consider these F_i as the basis and for 80% middle of F_i, the same as before, we replace each F_i separately with the other two F_i. The polynomials are computed, and if the difference was lower than Benford values, we will continue with this polynomial. In this case, starting at the beginning or end of the 80% middle of F_i for checking can be randomly.

As mentioned in the previous hybrid method, usually all approximate algorithms that compute F_i are executed in a certain number, their values are averaged together, and after rounding, the final F_i is put in the main function to compute the polynomials.

In the hybrid approach, it is assumed that the F_i of each method are given into the integration algorithm as inputs after a number of runs (e.g., 1000 times). For this reason, in the integration approach, practically the three categories of F_i are the input of algorithm and its output is an optimal polynomial.

The significant point is that one of these three categories of F_i may be a category that has been derived from the integration of the other proposed methods with this approach. In other words, the proposed integration scheme can be applied and presented in future research. In the next section, we will show for some given graphs how the results of implementing the proposed method to integrate the three categories of F_i will improve the final reliability polynomial.

3.3 An efficient numerical interpolation technique

In interpolation techniques, many points are extracted from the reliability polynomial graph. Then, with utilizing such points, the best polynomial to fit these points is extracted. An approach, which the equation of a graph can be obtained based on a series of points on the graph, is called interpolation.

In this section, for interpolation of these points, the Monte Carlo method is utilized to find the number of necessary and sufficient points of reliability polynomials. Further, the Legendre polynomial method, which is a powerful scheme for interpolation of points, is used. In what follows, we first need to briefly explain each of these concepts and then come up with the proposed method.

The crude Monte Carlo method

To obtain the crude Monte Carlo equation, it is assumed that with a certain probability, such as p, we have to find the connectivity probability of graph G. The crude Monte Carlo method (CMC) simulates an N sample $\varphi_1, \ldots, \varphi_N$ of i.i.d random variables with respect to the network structural functionφ. Based on the arithmetic mean of the N samples and according to the rule of large numbers, the estimator $\overline{\varphi}$ is presented for $R(G,p) = P\{\varphi = 1\}$ while the estimator is unbiased and $E[\overline{\varphi}] = R(G,p)$. According to the central limit theorem (CLT), the confidence interval for R (G, p) can be determined.

In general, the Monte Carlo method as a numerical solution is often used for the problems whose processes are random, discrete, and time-invariant, and their analytical solutions are highly complicated. In this scheme, for every probability between zero, one, a failure for the edge in the network (with a specified accuracy for example one thousandth) and a large number of times (e.g., 1000 times), the network edges are removed with a given probability and then check whether the final graph is integrated or disintegrated. Finally, the percentage of times the connected graph is computed helped us to obtain a value between 0 and 1 for the probability of failure. In fact, the number of these regular pairs is some points on the reliability polynomial's diagram.

In this article, we address the calculation of coefficients of the reliability polynomial by an interpolation technique, which computes such coefficients in a polytime with the size of graph accurately. Modified Newton's interpolation for estimating the polynomial on finite sets $\{p_i\} \in [0, 1]$ and the least squares error techniques are fortified with together to find a feasible reliability polynomial of interpolation technique.

3.3.1 Interpolating the coefficients of reliability polynomial

Orthogonal polynomials, as a collocation method, play an important and vital role in many spectral methods. One of the special series of orthogonal polynomials is Legendre polynomials, which is the answer to the following differential equation.

$$(1 - x^2)\frac{d^2y}{dx^2} - 2\frac{dy}{dx} + n(n+1)y = 0 \tag{17}$$

The n Legendre polynomials are obtained by the following recursive equation.

$$\begin{aligned} &P_0(x) = 1,\ P_1(x) = x, \\ &P_{n+1}(x) = \left(\frac{2n+1}{n+1}\right)xP_n(x) - \left(\frac{n}{n+1}\right)P_{n-1}(x),\ n \geq 1 \end{aligned} \tag{18}$$

Legendre polynomials are orthogonal on $[-1, 1]$ with weight function $w(x) = 1$.

$$\int_{-1}^{1} P_n(x)P_m(x) = \frac{3}{2n+1}\delta_{m,n} \tag{19}$$

in which,$\delta_{m,n}$ denotes Kronecker delta function. In what follows, we discuss the interpolation with Legendre polynomials.

Based on the explanations given for interpolation, it is known that any continuous function in the range $[a, b]$ can be approximated as:

$$f(x) = \sum_{i=0}^{\infty} a_i x^i \tag{20}$$

Since the Legendre polynomials in $[a, b]$ are complete, then $f(x)$ can be approximated as:

$$f(x) = \sum_{i=0}^{\infty} b_i P_i(x) \tag{21}$$

The reliability polynomial function (i.e., R) is a polynomial of maximum n (the number of graph nodes). Hence, we get

$$R = \sum_{i=0}^{\infty} b_i P_i(x) \tag{22}$$

where P_i are Legendre polynomials that are recursively derived from Eq. (18). For this reason, if one can define b_i's in the above equation, then the reliability polynomials can be obtained. The following equation can be used to extract b_i quantities.

$$\begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} P_0(x_0) & P_1(x_0) & \cdots & P_n(x_0) \\ P_0(x_1) & P_1(x_1) & \cdots & P_n(x_1) \\ \vdots & & \ddots & \vdots \\ P_0(x_n) & P_1(x_n) & \cdots & P_n(x_n) \end{pmatrix}^{-1} \begin{pmatrix} R(x_0) \\ R(x_1) \\ \vdots \\ R(x_n) \end{pmatrix} \tag{23}$$

in which x_i's are called the interpolation points. The quantities x_i's are the values of the interpolated x and R (x_i) are the values of the interpolated y. Thus, if we obtain n interpolated points (n is not the number of graph nodes) using the Monte Carlo method and then put them in the above equation, the values of b_i are achieved. If such b_i's are replaced with those P_i's calculated from Eq. (18), then the reliability polynomial can be interpolated effectively.

3.3.1.1 Interpolating the coefficients using legendre polynomials: a proposed approach First, we generate n points by the CMC method. Then interpolation process is carried out with Legendre polynomials, which is $a_0 + a_1 x + ... + a_n x^n$ a form of interpolation. It may be in the form of the extremum of a function that is why we derive the resulting polynomials, find the extremum points, and then apply the CMC method again for those points.

In the next step, we fit the polynomials $a_m x^m + \cdots + a_n x^n$ to these points as a fitting curve. Then we make derivation again and repeat it until the fitting curve is strictly increasing (decreasing) and has no root of zero. In the next step, the correction process of the coefficients is performed using the modified Newton's

Algorithm: The Modified Newton Method

1. $x_0 = \lceil a_{n-m} \rfloor, \cdots, \lceil a_n \rfloor^T$ // $(a_{n-m}, \ldots, a_n)$ is an input
2. $f[i] \leftarrow \int_0^{t_i} (R_f - R_I)^2 dx$
3. $J[i,j] \leftarrow \dfrac{\partial f[i]}{\partial a'[j]}$
4. $a'[j] \leftarrow \lceil x_0[j] \rfloor$
5. err $\leftarrow 1$
6. **for** $k = 1, 2, 3, \ldots$ **do**
7. $\quad x_0 \leftarrow x_0 - J^{-1}.f$
8. $\quad a'[j] \leftarrow \lceil x_0[j] \rfloor$
9. $\quad$ **if** $\int_0^1 (R_f - R_I)^2 dx <$ err **then**
10. $\quad\quad$ err $\leftarrow \int_0^1 (R_f - R_I)^2 dx$
11. $\quad\quad$ ans $\leftarrow a'$
12. $\quad$ **Endif**
13. **Endfor**

Fig. 2 Pseudo-code of the proposed interpolation algorithm using Legendre polynomials

algorithm, which is shown with details in Fig. 2. The output of the algorithm will be the desired polynomial coefficients with the desired precision. Figure 2 reveals the pseudo-code of the modified Newtonian algorithm.

The algorithm works by taking a sequence of real numbers as input data and then using the integer part of these numbers as a starting point. In this algorithm, the symbol $\lceil \cdot \rfloor$ represents the rounding operator, which means rounding the vector, i.e., rounding each of its components. In the second line, f is a vector whose elements are provided by values of Eqs. 1 to $n-m$, and thus, the construction of the desired function in this section is done. In the third line, the function f is minimized, and in Line 4 of the algorithm, $a'[j]$ is the coefficients of R_I where $m \leq j \leq n$. The goal of this algorithm is to obtain a polynomial that minimizes the mean-squared errors (MSEs) of the polynomials with the real coefficients $(R_f - R_I)$. For this purpose, Newton's method is used to solve the equations (Lines 6–8 of the pseudo-code shown in Fig. 2). Thus, the obtained function is determined each time in terms of MSE criterion. It should be noted that the meaning of J^{-1} in Line 7 is the inverse of the matrix J. Further, "ans" in Line 11 of the algorithm is the final solution vector. Moreover, in order to stop the algorithm, it is needed a stop condition that is considered by introducing "err" parameter, which means the quantity of error. Lines 6–10 form the body of an iterative loop, and the main purpose of this loop is to convert the minimum coefficients obtained by the previous steps into the integer coefficients as much as possible. In addition, it should be noted that in Line 7 of the algorithm, the equations formed in the previous steps are solved using the Newton–Raphson method.

According to the definition of the function R, which is a polynomial, the ultimate goal here is to minimize $A = \int_0^1 (R_f - R_I)^2 dx$ where R_f and R_I are float and integer functions, respectively. These functions can be expressed in one of two polynomial forms $a_0 + a_1 x + \ldots + a_n x^n$ or $a_m x^m + \cdots + a_n x^n$, $n > m$. It should be noted that these

polynomials expand the Legendre coefficients and then their integrals are calculated. The quantity A is a preliminary integral in terms of the polynomial functions and is therefore directly computable. If one is interested to minimize A, then it is necessary to minimize it for every $0 < t_i < 1$.

Hence, the main goal can be minimizing $A_i = \int_0^{t_i} (R_f - R_I)^2 dx$, which also implies zero derivatives related to the polynomial coefficients, so we get

$$\frac{\partial A_i}{\partial a_1} = \frac{\partial A_i}{\partial a_2} = \cdots \frac{\partial A_i}{\partial a_n} = 0 \tag{24}$$

where a_i are the interpolation function coefficients. Thus, in the proposed algorithm, the Jacobian matrix is constructed at first. In fact, there are n nonlinear equations here; the well-known algorithm for solving this nonlinear equation system is the Newton–Raphson algorithm that converts a nonlinear equation to linear one. However, it should be noted that the Obtained answers of the Newtonian's approach are real. Therefore, in order to obtain the integer solutions, a modification to the Newton's method has to be applied that these necessary modifications have been specified in Lines 9–11 of the pseudo-code in Fig. 2.

In addition, we also need an initial value (starting point), which comes from rounding the vector of the least squared errors. It is worth noting here that this starting point may not necessarily be the optimal answer, so we try to optimize the answer in the loop's body in Lines 6–13. Using the proposed algorithm, it does not need to find the number of spanning trees. The number of these trees is predictable with a good approximation, and it is the strength of the suggested algorithm. In the next section, according to the simulation results, we present how the implementation of the proposed methods in this paper leads us to achieve an optimal polynomial is close to the exact reliability polynomial of a given graph. In what follows, we estimate the computational complexity to assess the efficiency of the proposed interpolation method.

To calculate the integral, one of the available efficient methods can be used. In the present study, the numerical method proposed by Batra [33] has been utilized. According to [33], the computational complexity of integral in Line 2 has the execution order as $O(n^2 h \log n \log\log n)$ where h is the iteration step parameter. The execution order of Lines 3 and 4 is the constant as $O(1)$. Indeed, the execution order of Lines 1 to 5 does not play a fundamental role in calculating the complexity order. To determine the main complexity, it is necessary to proceed from the iteration loop (Lines 6 to 13) of the algorithm. The execution order of Lines 7 and 8 is $O(n^3)$ and $O(1)$, respectively. According to the study reported by Batra [33], the order of Line 9 is $O(n^2 h \log n \log\log n)$. The parameter k in Line 6 indicates the number of iterations in the loop body. Hence, the total complexity will be $O(k(n^2 h \log n \log\log n + n^3))$, which is approximated by $O(kn^3)$. Since the execution order of the integral is less than of the inverse matrix (J^{-1}), $O(n^3)$ is dominated and its effect will be ignored.

As mentioned, k is the controlling parameter of the body of the iteration loop and takes the integer values 1, 2, and so on. Even if k is the form of order n, it will be n^4 at worst, and if it is the form of n^2, it will be n^5 at worst. Thus, it is certainly not a kind of the exponential complexity order. The time complexity of the interpolation method

suggested by Robledo et al. [9] for the polynomials with the real coefficients has a uniform error boundary, and its complexity is reported as $O(2^m/\sqrt{N})$, where m is the order of the graph (the number of edges) and N is the sampling size in CMC method. Obviously, by increasing the sampling size, the error and consequently the complexity of the interpolation method can be reduced. Due to the increase in the order and size of the graphs in real networks, Robledo's scheme [9] has more complicated calculations; that is, the estimation of the reliability polynomial coefficients will be the form of the exponential order. Consequently, the interpolation method proposed in this study is more efficient and has a better degree of the complexity order.

4 Experimental results

Nowadays, computer systems are becoming much more complex and growing rapidly. The result of this growth is the increasing necessity of tools and techniques to identify behaviors of systems accurately such as performance and cost over time, system life, and so on. Simulation and mathematical modeling are two major ways to achieve these goals.

The performance evaluation of a complex network based on the reliability measurement can be a valuable measure. Since the measurement of reliability on the complex networks is usually difficult, mathematical and simulation models as two ways in designing and predicting the network behavior have been considered. Hence, the analytical models are expanding for describing the complex network behavior. Since the mathematical models cannot fully describe the behavior of a network, the outputs of these models are compared with the actual values measured from the network to validate the model. If the difference is within an acceptable range, it can be claimed that the presented analytical model is valid.

In Sect. 3, four approaches are discussed. In the current section, implementation, analyzing and evaluating of such methods are presented. The proposed schemes in the previous sections are summarized as follows.

1. Estimation of F_i using a proposed iterative method
2. Smart integration of F_i values extracted from top-down, bottom-up, and iterative methods
3. Integration of F_i values captured from top-down, bottom-up, and iterative methods using Benford's law
4. Obtaining the reliability polynomials using interpolation of Legendre polynomial

The validity of applying each of the above approaches with details is explained in the previous section. In this section, the simulation results obtained by implementing of mentioned approaches above on a set of graphs are presented.

4.1 Sample graphs

In the previous section, the importance of simulating the complex networks was emphasized. To evaluate each complex network using simulation, we need a set of graphs. These graphs can be divided into two types of application-driven (real) or synthetic. In the application-driven model, the resulting graph is related to a real complex network, whereas in the synthetic model, the resulting graph is constructed based on the possible parameters of the model.

From another perspective, the sample of graphs can be divided into two categories. In the first category, the reliability polynomial coefficients can be obtained exactly. In the second, those coefficients because of the large number of nodes and edges (increasing the graph size) are not computable practically and must be calculated approximately. In the following, the evaluation parameters will be described, and we will find that these parameters will vary depending on how accurately we can calculate the polynomial. In general, the reliability polynomials cannot be determined for large-size networks exactly, it is necessary to use the numerical values derived from Monte Carlo simulations. Hence, the evaluation parameters under these conditions will be different.

Figure 3 demonstrates the simulated graphs as the samples. All graphs are selected in the size for which the exact value of the reliability polynomial coefficients can be calculated. These twelve graphs are simulated to evaluate the performance and effectiveness of the proposed methods. In this set of graphs, the topologies such as semi-mesh (G_{11}), graphs with low degree nodes (G_5), graphs with high degree nodes (G_{10}), complete graph (G_8) and others have been simulated. In addition, the specific details of these graphs, including the number of nodes, the number of edges, and the maximum degree of the reliability polynomial for each of the graph samples, are listed in Table 2.

5 Performance evaluation

In this section, the various evaluation measures are examined. Obviously, all evaluation measures must determine the difference between the exact values of the reliability polynomial provided by the analytical model and the approximate values resulting from the simulation. If we have the exact value of the coefficients, then the difference between the two diagrams, i.e., the discrepancy between the actual and the approximate ones is determined as an evaluation metric. If the exact values of the coefficients are known, the following criteria can be utilized to quantify the accuracy of the proposed method.

- *MSE*[1] *Criterion*: Usually this criterion is used when trying to determine the difference between two graphs. In this criterion, it is assumed that the two graphs are plotted and the difference between them is to be calculated. To calculate

[1] Mean Squared Error.

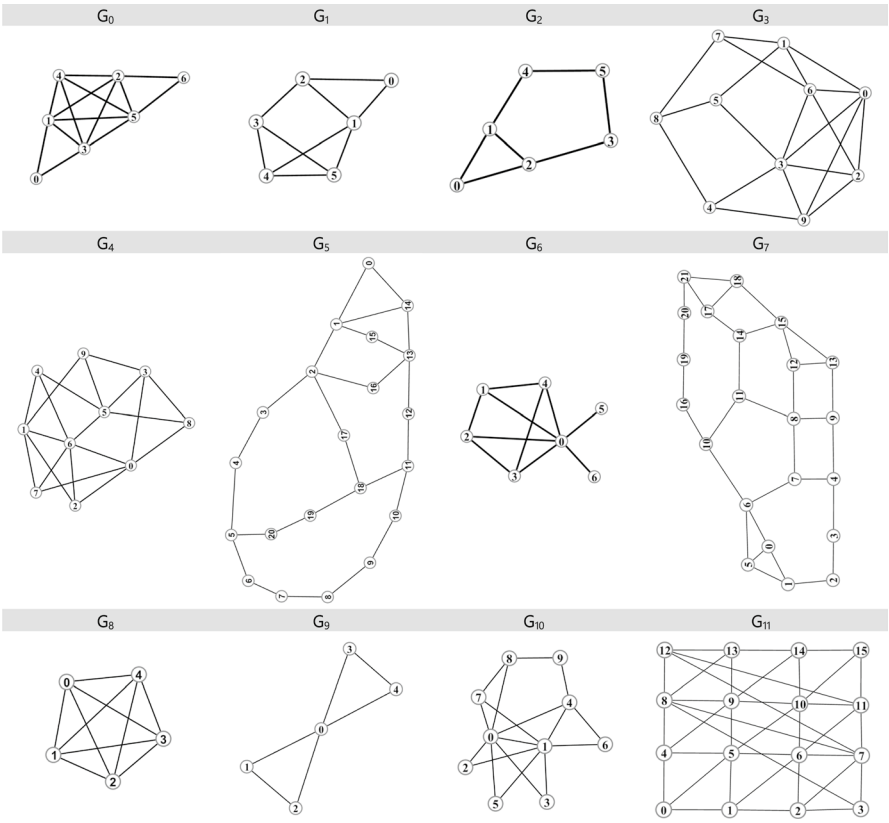

Fig. 3 The sample graphs used for different proposed methods

Table 2 Characteristics of the sample graphs illustrated in Fig. 3

Graph G	#Nodes	#Edges	Min-degree of R(G,p)	Graph G	#Nodes	#Edges	Min-degree of R(G,p)
G0	7	14	8	G6	7	10	4
G1	6	9	4	G7	22	32	11
G2	6	7	2	G8	5	10	6
G3	10	20	11	G9	5	6	2
G4	10	20	11	G10	10	17	8
G5	21	26	6	G11	16	24	9

the difference between the graphs, with predetermined precision, consider the points as *X*-axis of diagrams (the probability value of *p*). Then, these points are assigned to both graphs, and we obtain two values for *Y*-axis (i.e., the corresponding reliability value). These two values are subtracted from each other

and the power is doubled to eliminate the positive or negative issues. Then, these values for every X are averaged. Obviously, the higher the accuracy of the X (the distance between two X's from each other or their number in a fixed interval), the higher the accuracy of the MSE value. MSE criterion for the exact reliability polynomial (R) and the estimated reliability polynomial captured from a proposed approach (R') can be obtained from the following equation.

$$MSE = \frac{1}{n+1} \sum_{i} [R(i) - R'(i)]^2 \; i \in \{0, \frac{1}{n}, \frac{2}{n}, \frac{3}{n}, ..., 1\} \tag{25}$$

- *F_i-Difference Criterion*: Since the accurate and estimated values of F_i are available, the difference of F_i*'s* can be averaged. As mentioned earlier, the index i in F_i can range from 0 to $m-n+1$. This criterion is called E_{Fi}. For the two reliable polynomials (with F_i) and the estimated polynomials (with F'_i), this criterion is obtained by:

$$E_{F_i} = \frac{1}{m-n+2} \sum_{i=0}^{m-n+1} |F_i - F'_i| \tag{26}$$

- *The difference of the final coefficients of the reliability polynomials*: Another criterion that can be used to evaluate the proposed methods is to compare the coefficients of the two reliability polynomials after replacing the numerical values of F_i. In this criterion, like the previous criterion, the absolute value of the difference between the final reliability polynomial coefficients is calculated and finally the mean of them is obtained. The final coefficients of the reliability polynomials are also from p^0 to p^{m-n+1}. This is referred as E_{ci}.[2] For two exact reliability polynomials (with C_i coefficients) and estimated reliability polynomials (with C'_i), this criterion is given by:

$$E_{C_i} = \frac{1}{m-n+2} \sum_{i=0}^{m-n+1} \left|C_i - C'_i\right| \tag{27}$$

We mentioned earlier that if for any reason the exact value of the reliability polynomials coefficient is not available, then we need to use the values derived from the Monte Carlo method to perform the evaluation. The numerical values of Monte Carlo simulation are actually pointed on the graph. Since the exact values of F_i and C_i are not available in this case, we cannot use the E_{Fi} and E_{co}. In this case, the only evaluation criterion will be MSE. The Monte Carlo simulation output yields a pair of ordered X and Y, where Y as the exact output values of the reliability polynomial and X as the input values to the approximate polynomial. The MSE criterion for evaluating the estimated reliability polynomial (R') with n numerical pairs (x_i, y_i) derived from Monte Carlo simulation is given by:

[2] Error of coefficient.

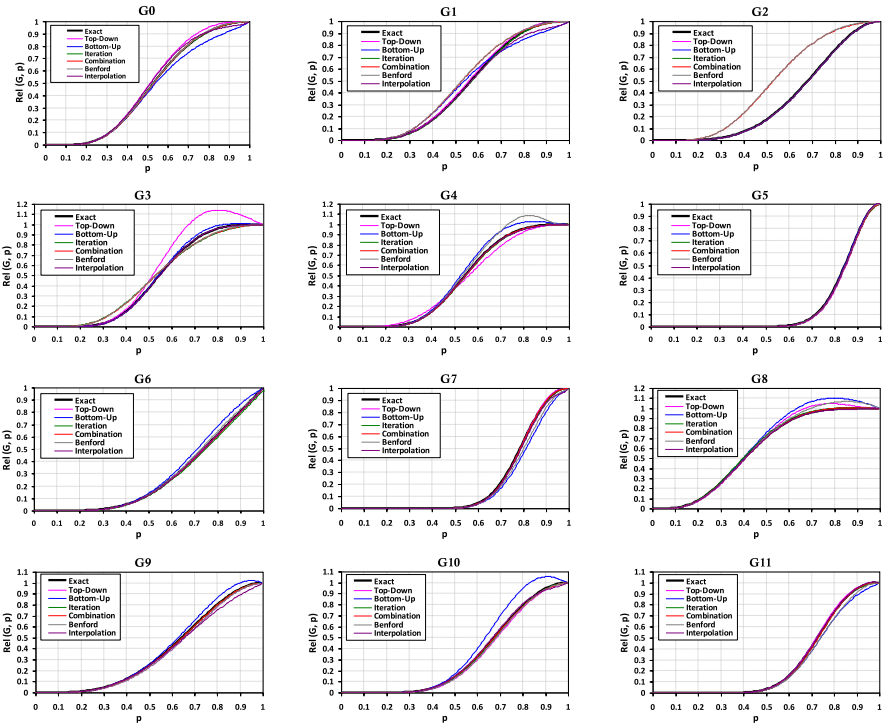

Fig. 4 Reliability polynomial diagrams, $Rel(G, p)$, for graphs G0 to G11 in terms of the probability of link failure (p) in Exact, Top-Down, Iteration, Combination (smart merging), and Interpolation methods

$$\text{MSE} = \frac{1}{n} \sum_{i \in x_i} [R'(x_i) - y_i]^2 \tag{28}$$

5.1 Analyzing the proposed methods

In this section, the proposed methods are simulated and evaluated. In order to do validation, the empirical experiments have been conducted using MATLAB due to the ease and efficiency of the matrix operation. First, the exact methods to determine the coefficients of reliability polynomial including top-down and bottom-up algorithms are implemented, and then, their performance will be evaluated for some given graphs. Then, all the methods are implemented separately and evaluated based on the evaluation metrics mentioned above.

Initially, for each of the twelve graphs (see Fig. 3), the reliability polynomial diagrams for all methods are plotted. The reason is to present a more comprehensive overview of each of these methods. In Fig. 4, the reliability polynomial diagrams are illustrated for graphs G0 to G11, respectively. In these diagrams, the

exact values obtained by the iterative method, the smart merging of coefficients, the integration approach using Benford's law, and finally, the proposed interpolation method with Legendre polynomials have been simulated and demonstrated.

Comparing the diagrams in Fig. 4 can yield very useful results on the different approaches that estimate the coefficients of reliability polynomials. The following is referred to them.

1. Graphically, (0, 0) is the starting point and (1, 1) is the end point as expected in all diagrams, but how to get this point is also important. For example, if we compare the two graphs G0 and G5, it can be seen that when p reaches 0.6, the reliability of the graph G5 is almost zero. However, the output value excludes of zero after passing p of 0.2 at G_0. This phenomenon suggests that G_5 graph is more tolerable than G_0.
2. In graph G2, F_i values are so small, which all the methods are fully able to approximate the coefficients as well as the exact reliability polynomials. It is observed that all diagrams are located on each other.
3. In some graphs, such as G_3 and G_4, the output of some of the estimated methods (available methods and not the proposed methods in this paper) exceeds the value of 1, which is false due to the normal reliability level that is between zero and one. This problem is rooted in the estimation method in some upward conditions.
4. In graph G9, which is a small size graph, although all methods are expected to calculate the reliability polynomial accurately, each of the two top-down and bottom-up algorithms for one of F_i had one error unit, and this error did not occur in the iterative method. It is worthy to note that in the smart integration method, we have finally been able to achieve the exact value, although we have used the values of F_i of all three top-down, bottom-up and iterative methods. That is why we call it the smart integration method because we know exactly which values of F_i in each method should we extract to be close to the exact value.
5. In graph G8, the reliability diagram for the two top-down and bottom-up methods exceeds 1. We have already stated the reason for this. The important point in this situation is that the method of integration by the Benord's law has not been adequately overcome this problem, while the smart integration method has been able to satisfactorily solve this problem, and it has not provided a value more than 1. Because it knows, exactly what values of F_i will take to bring the result closer to the exact value.

It is not clear from the diagrams of Fig. 4 that which method is better than the other methods. Hence, for a fair comparison it is recommended to utilize the evaluation criteria set out in Sect. 4. Therefore, the reliability polynomials of sample graphs are comparable. One criterion is MSE, in which two diagrams are compared point by point and the difference between them is measured. In Table 3, under the MSE criterion, the comparison values for each of the top-down and bottom-up methods as well as the four methods proposed in the manuscript with the exact value of the coefficients of reliability polynomial have been evaluated. As shown in the mean

Table 3 The values of MSE obtained by applying different methods

Graph	The method					
	Top-Down	Bottom-to-Top	Iteration	Smart Merging	Benford's law	Interpolation (Legend's Polynomial)
G0	0.018707	0.001278	1.11E-05	1.38E-05	2.62E-05	0.000432128
G1	0.000181	0.003381	4.81E-06	0.000106	0.000139	0.000332582
G2	0	0	0	0	0	0
G3	0.0118	0.000375	4.59E-05	0.000133	0.000361	1.13E-05
G4	0.000102	0.00212	8.82E-06	2.08E-06	0.00307	9.23E-06
G5	2.16E-05	4.28E-05	8.40E-07	7.54E-08	4.95E-06	1.77E-05
G6	8.31E-06	0.001394	3.40E-06	0	0.000212	2.83E-06
G7	5.02E-05	0.00161	3.85E-06	1.28E-05	0.000571	5.98E-05
G8	0.001106	0.004062	0.000206	1.54E-05	0.001212	5.15E-06
G9	0.000155	0.001164	0	0	0.000155	0.00062098
G10	0.000143	0.006375	2.63E-06	2.63E-05	5.76E-05	0.000191181
G11	0.000105	0.000749	8.04E-06	1.36E-05	0.000458	3.50E-05
Mean	0.002698	0.001879	2.46E-05	2.69E-05	0.000522	0.000143166
Normalized to the best result	109.5634	76.30813	1	1.092221	21.20915	5.8131709

row of the table, the mean of MSE for each of the twelve plots is calculated. Finally, at the last row, all normalized averages to the best value are expressed.

According to the normalized values, the iterative approach performed better than the other methods. With a very slight difference, the smart integration method comes in second. With a slight difference, third and fourth ranks are the interpolation approach (based on the Legendre polynomial) and the integration method (based on the Benford's law), respectively. The last two ranks belong to the top-down and bottom-up algorithms [21, 23], which are the old random algorithms. In summary, the results listed in Table 3 show that among all the methods, the efficiency and performance of the proposed methods were better than the two top-down and bottom-up algorithms.

Another evaluation parameter introduced in Sect. 4 is the difference values of F_i with their exact values. The values are depicted in Table 4. The last two rows of the table are similar to Table 3. According to this criterion, the smart integration approach gets the best rank, the second and the third ranks are assigned by iterative, and interpolation approaches, respectively. Moreover, the integration method (based on Benford's law) is worse than the bottom-up algorithm while almost the difference values for all three top-down, bottom-up, and Benford methods are close to each other. The worst-case of the integration method using Benford's law can have the same results as the worst-case in the three top-down, bottom-up and iterative algorithms. This criterion tries to have an equal and fair view of all the approaches. However, it is possible that it does not move in the right direction.

Table 4 The difference between the values of F_i captured by the proposed methods

Graph	The method					
	Top-Down	Bottom-to-top	Iteration	Smart merging	Benford's law	Interpolation (Legend's polynomial)
G0	72.444444	35.888889	18.6666667	12.5555556	26.4444444	69.3333333
G1	2	7.4	0.4	1.8	1.8	0.4
G2	0	0	0	0	0	0
G3	6588.9167	1430.6667	1150.33333	1922.41667	2965.91667	449.416667
G4	929.75	3920.5	688.833333	613.916667	2902.83333	559.75
G5	806.42857	49.857143	142.142857	26.8571429	643.142857	392.857143
G6	1.4	2.8	0.6	0	1.6	0.8
G7	239,121.17	211,302.92	64,697.4167	46,718.75	213,387.25	115,828.917
G8	9.1428571	7.1428571	4.57142857	1.28571429	4.71428571	2.57142857
G9	0.3333333	0.3333333	0	0	0.33333333	0.66666667
G10	170.66667	462.77778	44.8888889	124	133.666667	159
G11	2243	3639.1	3427.9	2694.1	3637.5	3540.3
Mean	20,828.771	18,404.949	5847.97943	4342.97348	18,642.1001	10,083.6677
Normalized to the best result	4.7959701	4.2378681	1.34653814	1	4.29247386	2.32183496

Table 5 Difference between the final coefficients obtained from the proposed methods

Graph	The method					
	Top-down	Bottom-to-top	Iteration	Smart merging	Benford's law	Interpolation (Legend's polynomial)
G0	6827.33333	235.8667	1106.6667	265.733	711.46667	1168.13333
G1	12.4	46.6	3.2	12	13.2	1.8
G2	0	0	0	0	0	0
G3	561,480.095	32,027.52	343,579.43	624,716	619,577.71	44,867.4286
G4	78,082.6667	101,831.4	360,365.33	33,153.2	161,572.19	30,985.7143
G5	7341.62963	155.5556	512.51852	28.6667	9685.3333	5463.03704
G6	7.27272727	4.181818	4.3636364	0	2.9090909	3.27272727
G7	22,918,587.8	5,915,185	3,511,588.1	9,325,555	46,449,984	25,325,758.5
G8	27.6363636	8.181818	34.909091	12.7273	86.727273	14.7272727
G9	0.57142857	0.285714	0	0	0.5714286	0.85714286
G10	614.444444	2839.444	1229.4444	5893.56	2389.1111	1796.11111
G11	11,208.48	74,048.56	376,988	61,626.2	74,045.92	450,690.64
Mean	1,965,349.19	510,531.9	382,950.99	837,605	3,943,172.4	2,155,062.52
Normalized to the best result	5.1321167	1.333152	1	2.18724	10.296807	5.62751516

The third and the last criterion against which the methods can be compared, is the difference criterion for the final coefficients of the reliability polynomial. The numerical values shown in Table 5 have compared the different methods from this point of view. According to this criterion, the iterative method has the best ranking. It is interesting that the second rank belongs to the bottom-up algorithm. It should be acknowledged that the criterion for the difference of the final polynomial coefficients is not at all a good measure since it does not actually consider the position of the coefficient in the reliability polynomials and what the coefficient of p is, and all coefficients are viewed with similar insight.

It is almost possible to say that the best evaluation criterion among these three criteria is MSE, which compares equitably and point by point the difference between the final diagram of the proposed method and the value of the reliability polynomial diagram. Putting this assumption based on the comparison, then it can be argued that all of the four proposed methods perform better than the two old top-down and bottom-up algorithms. Based on existing simulation experiments and the MSE criteria, the mean error for the two top-down and bottom-up algorithms was 27.1% and 18.8%, respectively. This value was reduced to 2%, 3%, 5.2%, and 14%, in iterative, intelligent combination, the combination based on Benford's law and the interpolation methods, respectively.

Another important point is that with the increase in the size of the graphs, it will be no longer seen this error reduction process. In other words, for large graphs, the percentage error reduction will not be linear, and it cannot be argued with the proportionality ratio that if the error of the top-down algorithm is reduced from 27.1% to 27%, then the error of the iterative method will remain 20%. Nevertheless, clearly none of the four proposed approaches in the present study increases the time complexity of the computation and thus improves the accuracy of the estimation to a much greater degree than the real state.

6 Robustness analysis

Robustness in complex networks is one of the vital and important features and has become one of the important and growing research fields in recent years. So far, various criteria have been proposed to measure and evaluate the strength and flexibility of networks that an important part of them is dedicated to the graph theory and its study.

Fault tolerance is a key consequence for evaluating and analyzing a network. Nodes and edges may be removed from the network due to faults, and as a result, the local or global characteristics of the network as well as its proper operation may change. Failures may not only reduce the operational and computational power of the network but may also alter the topology of the network and lead to its disconnection.

Extensive researches have been attempted in the related studies to evaluate the robustness of graphs and networks in order to explain the robustness of graphs against random failures and systematic attacks. In this section, we aim to introduce some of the most important of these measures and show how they can be interpreted

to assess the robustness of graphs. It is worth mentioning that our purpose is not about providing a comprehensive, completing research report on the types of robustness metrics and comparing their effectiveness. Appropriate classification of robustness metrics and finding similarities or inconsistencies between them is one of the esteemed research aspects in this field. However, the multiplicity and diversity of criteria for evaluating resilience and sometimes contradictions and inconsistencies in their interpretation may cause confusion for any researcher in explaining each of these measures. Therefore, accurate, appropriate and consistent classification of metrics, finding similarities and differences between them, as well as expressing the advantages and disadvantages of each and, most importantly, knowing a class of graphs that these measures can be used more reliable in them, can be considered as valuable and separate research work, solely.

6.1 Classifying the robustness metrics

For selecting and categorizing the robustness measures, one can rely on the literature review and a wide range of related keywords. There are a number of robustness measures that some of them share similar characteristics. A basic question is "how the robustness measures can be assigned to the appropriate categories without overlapping?" By studying the properties of graphs that represent a particular subject, as well as the networks which can be generated by models, the difficulties ahead can be largely overcome. In summary, the following three main factors can be considered for selecting an appropriate measure.

a. Ability to apply the desired criterion to at least one type of graph
b. Direct measurement of one of the known aspects of the robustness graph
c. A clear relationship between the selected robustness criterion and one of the known characteristics of the graph

In addition, the characteristics considered for selecting the robustness metric can be summarized in the following five cases.

1. Graph type (i.e., simple, weighted, un-weighted, directed, or in-directed)
2. Being globally or locally (focusing on the connectivity of each node or extracting properties about the whole graph)
3. Static or dynamic network under evaluation
4. Co-domain of the robustness measure
5. Ability to calculate the robustness measure, which may be NP-hard or NP-complete.

Thus, in the classification of graph robustness metrics, the following three major classes can be mentioned.

1. Distance-based metrics
2. Min-cut based metrics

3. Spectral-based metrics

In the continuation of this section, we will analyze each of these categories and one of the most important measures of each category as a detailed example. It should be noted that each metric may belong to one or even more main classes and, in other words, the classes overlap with each other. Obviously, a good classification method should be able to reduce this overlap as much as possible. This important issue needs to be addressed in future studies.

6.1.1 Distance-based measures

Any measure in this class answers the important question of how the path length between two nodes can be related to the network robustness. Random failures and targeted attacks increase the distance between nodes, reduce the speed of data transmission and increase the costs. For example, a node that is on average less distant from other network nodes can act as an important central node with high traffic throughput.

6.1.1.1 Diameter stability: persistence For finite paths, the co-domain of diameter is in the range $[1, n-1]$; otherwise, it is in the range $[1,\infty)$.

Definition 13 [34]: *Diameter stability is the minimum number of nodes that must be failed to increase the diameter of the network.*

This measure is important in graphs and communication networks because it shows how the latency of messages in the network may increase based on the failure of its nodes or edges. Usually, the vector called the persistence vector is defined as follows

$$P(G) = (p_1, ..., p_n) \tag{29}$$

where p_i represents the worst case of the diameter of graph G when i is the number of faulty edges (nodes).

6.1.2 Min-cut-based measures

The major issue in defining the classical connectivity is that it does not provide any information about which links or nodes are sensitive (critical) and it only refers to the vulnerability of networks. Hence, the metrics based on the min-cut can be seen as a derivation of the classical connectivity problem. Another problem is that the connectivity does not provide any information about the components and subnets that remain after the disintegration of the graph and does not evaluate the performance degradation by the failure of independent nodes and links. In the measures based on the min-cut, the main problem is to find the minimum number of components of the network, which their elimination will cause the disintegration of the

network. Sometimes the problem is NP-complete, especially for graphs with large sizes.

6.1.2.1 Toughness This measure indicates the survivability of a network, which was introduced by Chvátal [35]. This metric seeks to find how many connected components in the underlying graph can be partitioned upon a failure occurs in a certain number of nodes. In the graph theory, toughness can be a measure of the connectivity of a given graph G.

Definition 14 [35]: *For a given real number t, graph G is said to be t-tough if, for every integer k > 1, G cannot be decomposed into k different connected components by removal of less than tk vertices.*

Mathematically, the toughness of G is the largest t for which the following inequality holds

$$t \leq \frac{\|S\|}{\|G \backslash S\|} \tag{30}$$

in which S is an arbitrary set of nodes and $\|S\|$ is its cardinality. Moreover, $\|G\backslash S\|$ indicates the number of connected components remains in G after removal (failure) of the nodes of set S.

If the number of components obtained by removing a set of vertices is always greater than the number of vertices removed, then the toughness of graph is 1. In addition, it is proved that the lower limit of toughness is k_v/β_0 where k_v is the node connectivity and β_0 denotes the independent number of graph. This parameter indicates the size of the maximum set of nodes in which no two nodes in this set are directly connected to each other by an edge. This measure does not impose any constraints on the relative sizes of the components that remain as soon as the graph is disconnected. However, it is not a suitable practical measure for assessing the network reliability. This can be the disadvantage of the toughness metric.

6.1.3 Spectral-based measures

Nowadays, an important part of algebraic graph theory is devoted to the study of the spectrum of an adjacent matrix of a graph, also known as a graph spectrum. By definition, the graph spectrum is matrices which uniquely represent the structure of a graph. For example, the spectrum of the adjacent, Laplacian, and the distance matrices.

6.1.3.1 Algebraic connectivity Definition 15 [36]: *Algebraic connectivity is the second smallest value of the Laplacian matrix L of a graph. Eigenvalues of L can be ordered such that* $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. *Thus, the algebraic connectivity is equal to* λ_2.

Algebraic connectivity can be considered as an instance of the graph connectivity and therefore a criterion for measuring the graph vulnerability. If and only if the graph is disconnected, then $\lambda_2 = 0$; it will be equal to the total number of vertices of the graph when all nodes are connected. According to Fiedler's theorem [36], the algebraic connectivity of any incomplete graph will not be greater than its connectivity of the vertex and edge. In addition, the multiplicity of roots of the values aequals zero corresponds to the number of connected components of the graph. Hence, it seems that the larger the measure of algebraic connectivity, the greater the robustness of the graph. This metric is global and static that can be easily computed. It should be noted that the algebraic connectivity depends only on topology and is a de-normalized measure. Another disadvantage of this metric is that for a disconnected network, its value is zero, while there may still be sufficiently large connected components remaining in the network.

6.1.3.2 Number of spanning trees A spanning tree is a subgraph that contains $n-1$ edges which contain all n vertices without any cycle. However, this definition is not unique; that is, a graph can have several spanning trees. One way for evaluating the robustness is counting all possible spanning trees for a given graph. The result is normally an integer that can grow exponentially. Baras and Hovarshti [37] have shown that the number of spanning trees in the graph abstraction of the communication system could be used as a reasonable measure for system robustness against the communication topology changes.

According to Kirchhoff's tree matrix theorem [28], all cofactors of Laplacian matrix of graph G are equal to the number of spanning trees. In the ATR-reliability problem, the number of min-paths is the same as the number of spanning trees.

A very interesting point to note here is that the reliability polynomial of a graph has a positive correlation to the number of spanning trees of that graph. It is shown that the reliability polynomial of graph G, when $p \longrightarrow 0$, denotes the following equation [27]

$$R(G) = \xi(G)p^{n-1} + O\left(p^{n-1}\right) \tag{31}$$

in which $\xi(G)$ indicates the number of spanning trees of graph G. Interestingly, this metric makes a similar decision with the reliability polynomials (when $p \to 0$) about comparing the robustness of two arbitrary graphs G1 and G2. Mathematically, we get

$$if \xi\left(G_1\right) < \xi\left(G_2\right), p \to 0 \Rightarrow R\left(G_1\right) < R\left(G_2\right) \tag{32}$$

In this manner, the reliability of the two graphs can be compared with each other through the number of their spanning trees. It should be noted that in real networks, the edge failures are rare and therefore the assumption of $p \longrightarrow 1$ is more reasonable. However, when p tends to zero, the number of spanning trees will correspond to the reliability polynomial.

In order to have a fair and accurate assessment of the robustness of graphs, the values of the independence number, toughness, algebraic connectivity, and the number of spanning trees for the graphs G0 to G11 (see Fig. 3) are determined and listed

Table 6 Numerical values of independence number, toughness, algebraic connectivity, and the number of spanning trees for graphs G0 to G11 shown in Fig. 3

Graph	Independence No. (β0)	Toughness (t)	Algebraic connectivity (λ2)	No. of spanning trees
G0	2	2.5	1.55	720
G1	2	2	1.43	64
G2	2	2	1	14
G3	3	2.33	1.61	24,704
G4	5	1	1.65	24,693
G5	7	2	0.17	15,667
G6	4	0.75	1	45
G7	8	1.75	0.21	2,059,264
G8	1	4	5	125
G9	2	1.5	1	9
G10	6	0.66	1.23	2252
G11	4	3	1.36	208,328,480

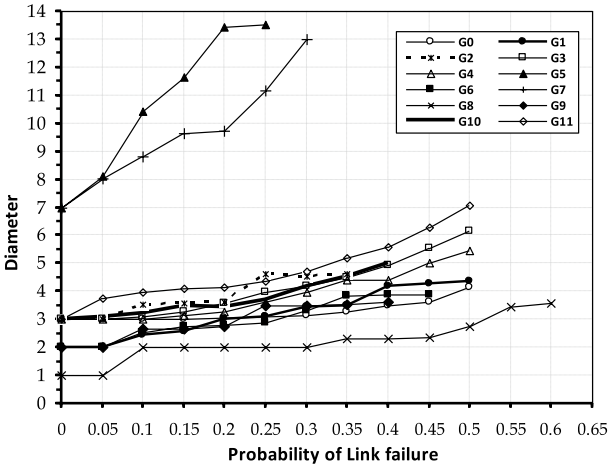

Fig. 5 Comparison the robustness of G_0 to G_{11} networks (shown in Fig. 3) based on the diameter persistence (stability) in terms of the probability of link failure

in Table 6. With these numerical quantities, in addition to the reliability polynomials, a fair comparison of graphs based on their robustness can be achieved.

As mentioned before, to measure the robustness of graphs, suitable metrics are needed to capture the network capabilities adequately, including graceful degradation, against all types of failures. Metrics such as the connectivity are usually worst-case criteria and have limited expressiveness. So, in another step, we looked for more expressiveness measures, some of which were mentioned above, and listed their quantitative values in Table 6. Another important metric is the diameter

persistence (stability). According to this measure, one can evaluate the networks depicted in Fig. 3 and compare their resilience with each other.

Figure 5 reveals the comparison of the diameter persistence of graphs G0 to G11 (see Fig. 3). The horizontal axis in this figure reveals the probability of link failure, while the vertical axis depicts the variations in the network diameter due to such failures. The smaller the variations, the greater the diameter persistence. For example, the better grade belongs to the complete graph G8, and the graphs such as G_5 and G_7, which correspond to real-world networks, have the lowest rank among the networks and have the lower diameter stability.

7 Concluding remarks and future work

The robustness of any network is usually expressed in terms of its tolerance to random failures and targeted attacks. Networks are highly subject to such treatments, and thus, evaluating and improving the resilience of networks are essential aspects of their design. Deciding whether a network is robust requires a method of quantitative measurement. Over the recent years, various metrics have been proposed to measure the network's robustness. However, scientists working in this field have not reached an agreement on which measures are superior to the other; so, this need and effort is still ongoing. In this article, we briefly review the network robustness measures and three important classes, i.e., connectivity-based, distance-based, and spectral-based classes. We aimed to suggest efficient methods to explain the degree of resilience of networks. The main purpose of this study was not to provide a comprehensive report about such metrics and compare their effectiveness thoroughly. Proper classification of measures and finding similarities or inconsistencies between them is one of the aspects of valuable research that is suggested as one of the important future work to continue this research.

The reliability polynomial of a graph is one of the most important criteria for measuring the robustness of graphs based on the graph spectrum. Unlike the other metrics that attribute a numerical value (usually normalized) to the robustness of a graph, the reliability polynomial attributes a function in terms of the probability of link failure. Some of the practical applications of this research are array-based designs (such as vertical FET (VFET), vertical slit FET (VeSFET), FinFETs and nano-electromechanical systems (NEMS)), the arrays of CMOS, an axon communication, probabilistic gate matrix (PGM), molecular networks of biological systems, the effect of structural motifs on the diffusive dynamics (such as the spread of infectious disease), and finally, targeting interventions to control the outbreak of epidemics. These are all examples of applications that can be well-fitted to the networks whose robustness can be effectively determined and measured with the reliability polynomials.

One of the most important challenges for utilizing the reliability polynomial as a robustness indicator is determining its coefficients that can only be accurately determined for graphs of small order and size. Hence, the exact determination of the coefficients lies in the category of NP-hard problems. In this paper, to meet such a challenge, four efficient methods are suggested. In the first method, using an iterative

method, the reliability polynomial coefficients are estimated. This method, on average, has the same accuracy as the two methods in the literature, namely top-down and bottom-up approaches. In addition, the idea of smart merging the coefficients is suggested for three methods, namely, top-down, bottom-up, and iterative. In this scheme, the values of the coefficients for the smaller, the larger, and the middle indices, have been estimated, respectively, using the top-down, the bottom-up, and the iterative approaches. The simulation experiments in most of the underlying graphs demonstrate that the smart merging and the iterative methods have the best efficiency and accuracy among the other ones. Additionally, as a third idea, we fortified Benford's law to integrate the coefficients in a proper manner. Moreover, as a final idea, we utilized the Legendre interpolation method to estimate the coefficients of reliability polynomial, accurately.

One of the future researches that can be mentioned in the current study is that the assumption of failure independence in different network components may seem unrealistic in some cases. For example, in geometric and geographic networks, such as transportation networks and optical telecommunication networks, which can affect a large geographical area, the failures are no longer independent. The joint probability of two or more nearby component failures has a significant correlation. Further, it has shown that the failure probability of some links depends on the length of the link, and in such cases, we are dealing with geographically correlated failures.

References

1. Di Caprio D et al (2021) A novel ant colony algorithm for solving shortest path problems with fuzzy arc weights. Alex Eng J. https://doi.org/10.1016/j.aej.2021.08.058
2. Sori AA et al (2021) Fuzzy constrained shortest path problem for location-based online services. Int J Uncertain Fuzz Knowl-Based Syst 29(2):231–248
3. Sori AA et al (2020) Elite artificial bees' colony algorithm to solve robot's fuzzy constrained routing problem. Comput Intell 36(2):659–681
4. Sori AA et al (2020) The fuzzy inference approach to solve multi-objective constrained shortest path problem. J Intell Fuzzy Syst 38(4):4711–4720
5. Y. Khorramzadeh, (2015) Network reliability: theory, estimation, and applications, Ph.D. dissertation, Virginia Tech
6. Moore EF, Shannon CE (1956) Reliable circuits using less reliable relays. J Franklin Inst 262(3):191–208
7. Cowell ER et al (2018) On the exact reliability enhancements of small hammock networks. IEEE Access 6:25411–25426
8. Rohatinovici NC, Proştean O, Balas VE (2018) On reliability of 3D hammock networks. In: IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI), pp 000149–000154
9. Robledo F et al (2013) A novel interpolation technique to address the Edge-Reliability problem. In: 5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), IEEE, pp 187–192
10. Burgos JM, Amoza FR (2016) Factorization of network reliability with perfect nodes I: introduction and statements. Discret Appl Math 198:82–90
11. Khorramzadeh Y et al (2015) Analyzing network reliability using structural motifs. Phys Rev E 91(4):042814
12. Eubank S, Youssef M, Khorramzadeh Y (2014) Using the network reliability polynomial to characterize and design networks. J Complex Netw 2(4):356–372

13. Brown JI, Cox D, Ehrenborg R (2014) The average reliability of a graph. Discret Appl Math 177:19–33
14. J.I. Brown and K. Dilcher, On the roots of strongly connected reliability polynomials, Networks: An International Journal, Vol. 54, No. 2, pp.108–116, 2009.
15. Brown JI, Colbourn CJ (1992) Roots of the reliability polynomials. SIAM J Discret Math 5(4):571–585
16. Royle G, Sokal AD (2004) The Brown-Colbourn conjecture on zeros of reliability polynomials is false. J Comb Theory Ser B 91(2):345–360
17. Brown JI et al (2014) Inflection points for network reliability. Telecommun Syst 56(1):79–84
18. Brown JI, Koç Y, Kooij RE (2011) Reliability polynomials crossing more than twice. In: 3rd IEEE International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp 1–6
19. Page LB, Perry JE (1994) Reliability polynomials and link importance in networks. IEEE Trans Reliab 43(1):51–58
20. Chen Y et al (2003) A note on reliability polynomials and link importance in networks. In: International Conference on Neural Networks and Signal Processing, Proceedings of the 2003, vol 2, pp 1674–1676, IEEE
21. Beichl I, Cloteaux B, Sullivan F (2010) An approximation algorithm for the coefficients of the reliability polynomial. Congr Numer 197:143–151
22. Colbourn CJ, Debroni BM, Myrvold WJ (1988) Estimating the coefficients of the reliability polynomial. Congr Numer 62:217–223
23. Harris DG, Sullivan F (2015) Sequential importance sampling algorithms for estimating the all-terminal reliability polynomial of sparse graphs. In: Approximation, Randomization, and Combinatorial Optimization, Algorithms and Techniques (APPROX/RANDOM 2015), Schloss Dagstuhl-Leibniz-Zentrum Fuer Informatik
24. Harris DG, Sullivan F, Beichl I (2011) Linear algebra and sequential importance sampling for network reliability. In: IEEE Proceedings of the 2011 Winter Simulation Conference (WSC), pp 3339–3347
25. Rausand M, Hoyland A (2003) System reliability theory: models, statistical methods, and applications. Wiley, London
26. Colbourn CJ (1987) The combinatorics of network reliability. Oxford University Press, New York
27. W. Ellens, (2011) Effective resistance and other graph measures for network robustness, MS thesis, Leiden University
28. G. Kirchhoff, (1847) Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird. Ann Phys, 148: 497–508
29. Satyanarayana A, Chang MK (1983) Network reliability and the factoring theorem. Networks 13(1):107–120
30. Benford F (1938) The law of anomalous numbers. Proc Am Philos Soc 78:551–572
31. L. Zhipeng, C. Lin, W. Huajia, (2004) Discussion on Benford's law and its applications, Cornell University Library
32. Hill TP (1995) A statistical derivation of the significant-digit law. Stat Sci 10(4):354–363
33. Batra P (2008) Newton's method and the computational complexity of the fundamental theorem of algebra. Electron Notes Theor Comput Sci 202:201–218
34. Choi M, Krishna CM (1989) On measures of vulnerability of interconnection networks. Microelectron Reliab 29(6):1011–1020
35. Chvátal V (1973) Tough graphs and Hamiltonian circuits. Discret Math 5(3):215–228
36. Fiedler M (1973) Algebraic connectivity of graphs. Czechoslov Math J 23(98):298–305
37. Baras JS, Hovareshti P (2009) Efficient and robust communication topologies for distributed decision making in networked systems. In: Proceedings of the 48th IEEE Conference on Decision and Control (CDC), pp 3751–3756

Authors and Affiliations

F. Safaei[1] · R. Akbar[1] · M. Moudi[2]

R. Akbar
r_akbar@sbu.ac.ir

M. Moudi
mmoudi@torbath.ac.ir

[1] Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran

[2] Department of Computer Engineering, University of Torbat Heydarieh, Torbat Heydarieh, Razavi Khorasan Province, Iran