



Universitat Rovira i Virgili (URV) y Universitat Oberta de Catalunya (UOC)

Master in Computational and Mathematical Engineering

Final Master Project (FMP)

A simulation-interpolation approach for approximating the univariate and bivariate reliability polynomial

Name of the Student: Carlos Eguía León

Name of the Tutor: Dr. Hebert Pérez-Rosés

Date of Delivery: 21 June 2020

Dr. Hebert Pérez-Rosés certifies that the student Carlos Eguía León has elaborated the work under his direction, and he authorizes the presentation of this memory for its evaluation.

Director's signature:



Lleida, 16-06-2020



This work is subject to a licence of Recognition-NonCommercial- NoDerivs 3.0 Creative Commons

INDEX CARD OF THE FINAL MASTER PROJECT

Title of the FMP:	<i>A simulation-interpolation approach for approximating the univariate and bivariate reliability polynomial</i>
Name of the author:	<i>Carlos Eguía León</i>
Name of the director:	Dr. Hebert Pérez-Rosés
Date of delivery (mm/aaaa):	06/2020
Degree:	Master in Computational and Mathematical Engineering
Area of the Final Work:	Network Science
Language of the work: (Spanish, English or Catalan)	English
Keywords: Maxim of 3 keywords (proposed by the students or in base to listings, thesaurus, etc., and validated by the director)	<ol style="list-style-type: none"> 1. Network science 2. Bivariate reliability polynomial 3. Interpolation
<p>Summary of the Work (maximum 250 words): <i>With the purpose, context of application, methodology, results and conclusions of the work.</i></p>	
<p>The purpose of this work is to provide an alternative approach to the exact computation of the coefficients of the reliability polynomial of a stochastic network with edge failure (under the all-terminal reliability condition), in the univariate and bivariate cases.</p> <p>This exact computation has driven most research efforts for the past few decades, mostly in the univariate case, while in the bivariate case there is still little literature.</p> <p>Starting with a brief review of basic concepts, I have tackled the univariate case where all edges have the same failure probability. The proposed method approximates the graph's reliability for different failure probabilities taken at Chebyshev points in $[0,1]$ via simulation of the graph's state at a given point in time. These values will then form a Vandermonde matrix, which we can use to interpolate the graph's reliability polynomial.</p> <p>The next step is to apply the same reasoning to the bivariate case, where edges can fail with one of two different probabilities. While the approach uses the same combination of simulation and interpolation, the second phase differs from the previous univariate case in its higher complexity and lesser guarantee of accuracy. Nevertheless, the results suggest that this approach can be performant.</p> <p>Finally, there are many more assumptions that we could consider in order to bring the studied networks closer to real-life, but this paper's scope has already been challenging with the computational limitations of a personal computer. The use of cyberinfrastructures like clusters would have helped extend our conclusions.</p>	

Abstract (in English, 250 words or less):

This paper aims to provide an alternative approach to the exact computation of the reliability polynomial of univariate and bivariate network reliability by applying simulation-interpolation based methods. This 2-phase combination consists in approximating firstly the values of a network's reliability for different states (i.e. different edge failure probabilities) through simulation techniques, and secondly to use this information as interpolation nodes to approximate the network's reliability as a continuous function. These results, although approximated, can then serve as decision-makers in many real-life scenarios.

We review some basic concepts for a later development of the proposed approach in instances of gradually increasing complexity.

Index

1. Introduction	1
2. Basic definitions and notations	3
3. Univariate case	3
3.1 The reliability polynomial	3
3.2 Polynomial interpolation of the reliability polynomial	5
3.3 Practical application	8
3.4 Quality assessment of the proposed method and calibration	9
4. Bivariate case	10
4.1 The reliability polynomial	10
4.2 Polynomial interpolation of the bivariate reliability polynomial	13
4.2.1 Bidimensional univariate polynomial interpolation (BUPI)	13
4.2.2 Newton's bivariate polynomial interpolation (NBPI) in a rectangular basis	15
4.3 Practical application	17
4.4 Quality assessment of the proposed methods and calibration	20
5. Conclusions and future work	22
6. Bibliography	23
7. Annexes	24

List of figures

Figure 1: A simplified roadmap of Ibiza	2
Figure 2: Diagram of the proposed method for the univariate case	7
Figure 3: The graph $K_4 - \{e\}$	8
Figure 4: Numerical performance of the proposed method for the univariate case	9
Figure 5: Graphical performance of the proposed method for the univariate case	10
Figure 6: Matrix of the coefficients of the expanded form	14
Figure 7: Minimal coefficient matrix	14
Figure 8: Two sets of equations after a bidimensional univariate interpolation	14
Figure 9: Diagram of the proposed method: BUPI	15
Figure 10: Diagram of the proposed method: NBPI	17
Figure 11: Example scheme of interpolation points for the BUPI (filled in red) and the NBPI (all)	18
Figure 12: Bivariate $K_4 - \{e\}$	18
Figure 13: A simplified road map of Ibiza (with two types of roads)	19
Figure 14: Numerical performance of the proposed methods for the bivariate case	20
Figure 15: Graphical performance of the proposed methods for the bivariate case	21

1. Introduction

Networks surround us with all sorts of shapes, scales, and sizes, as the natural form in which groups of elements of any type interact with one another. From the study of a brain's interconnected neurons to the optimization of worldwide transportation systems, network science plays an increasingly important role in all sorts of areas. The need to better understand networks is an issue that multiple branches of knowledge, from biology to social sciences, are tackling through network science.

The urge to develop this relatively new discipline is significant, to the extent that the impact of networks in our lives is critical. If we think of how an infectious disease is transmitted within a network of individuals, understanding its spread through the population would guarantee that the right measures are taken against a risk of an epidemic. On a bigger scale, transportation systems are a real challenge that society has faced for centuries. Its social and economic impact, and more recently its extremely serious environmental consequences, have driven the search of the most effective ways to transport goods or persons between cities.

At the same time, the possibilities of development are enormous. Although in the past decades a significant number of studies have focused on relatively simple networks, increasing groups of researchers are exploring a wider and darker space that is bringing network science closer to real-life. We almost immediately think of a brain when considering complexity, but we may still be far from this level of intricacy. Complexity can also involve considering numerous factors where previous scientists have historically considered a single standardized variable. For instance, let us go back to the infectious disease problem. What if we considered two types of relationships between the group of individuals (those that imply an everyday contact -family, colleagues-, and those involving only a sporadic contact) instead of a single, plain type of relationship? This perspective would certainly be more realistic, and therefore would elevate the conclusions of network science to a more refined and useful tool to help make the world better.

In network science, one of the major areas of interest when analysing existing networks is to evaluate their reliability, or capacity to remain operational. Reliability can take different forms depending on what is the network's purpose. In the case of a transportation system, there may be a depot and a terminus station, and thus the system will be operational if there is a route between both stations. However, if the need is to ensure that there is always a route between any pair of stations in the network (like in a road network where no city should be isolated), then reliability takes on a different dimension. The study of network reliability as we know it today was first introduced by Shannon and Moore in the 1950's, and in their ground-breaking paper [4] they set the basis of a promising field of research where networks take the shape of a probabilistic model, and reliability is translated into a polynomial. This model's potential is immense: if we can numerically determine the reliability of a system, then for instance we could forecast to what extent an electric grid will remain operational under natural disaster conditions. Still there are many unknown details in network reliability, and in some cases their complexity renders their analysis intractable. In fact, we may be able to translate a system's reliability into a mathematical formula, but this formula remains partially symbolic. Moreover, when increasingly complex networks are being considered, the number of unknowns that arise when trying to determine the system's reliability multiplies.

However, beyond the bounds of exact computation, the theory of approximation can take over and help us nail down the work of previous network scientists. Approximation techniques have proven to provide excellent results in those cases where computational evaluation is too costly or just too complex. One approximation technique that has been widely studied is polynomial interpolation. In this paper we will use interpolation methods to try to fill in the gaps in the study of reliability of simpler networks, and then we will attempt to apply these same methods to more complex networks that can be shaped as bivariate probabilistic models. We will also combine this approach with simulation techniques to build it on sufficient numerical information.

Particularly, we will focus on systems that are stochastic networks with edge failure, and where reliability is defined as the capacity of any pair of elements in the network to remain connected (called all-terminal reliability). There are many examples in real-life of such systems. Let us consider for instance a road network within a given region, containing sparsely distributed cities, villages and road junctions interconnected by roads, like the one in figure 1. Any road section could be blocked due to accidents or natural events (landslides, floods, ...), but the network must guarantee that a vehicle departing from any point could get to any other given point in the map.

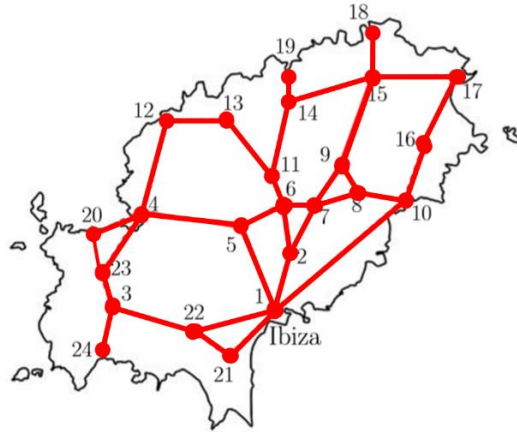


Figure 1: A simplified roadmap of Ibiza

We will start by briefly presenting the basic definitions and notations that we will use in this paper. Then, we will present our simulation-interpolation approach to the approximation of a network's reliability for the simplest case where edges can fail with the same probability (univariate case). After the results of this proposed method are shown, we will move to the second simplest case, the bivariate case: following our same approach, we will propose two approximation methods that will only differ in the interpolation technique. Finally, we will summarize our conclusions and discuss the future work.

2. Basic definitions and notations

In order to model a system, network science utilizes concepts from graph theory. Let us consider a network modelled by a graph $G = (V, E)$ where V is a set of n vertices or nodes, and $E \subseteq \binom{V}{2}$ is a set of m bidirectional edges or links (hence our graph is undirected). This network is considered to be operational if it satisfies all-terminal reliability.

A **pathset** is a subset $O \subseteq E$ of edges that makes the graph operational, so any connected spanning subgraph of G is a pathset. A minimal pathset with respect to inclusion is a **minpath**, so any spanning tree is a minpath. Similarly, a **cutset** is a subset $C \subseteq E$ of edges such that $(V, E - C)$ is not operational, and a minimal cutset is called a **mincut**.

Regarding edge failure, we will emulate the model introduced by Moore and Shannon where the nodes are considered to be perfectly reliable, and the edges could fail independently with a certain probability (this model is aligned with the examples of road networks, electric grids, or epidemic spread). At any given time, each edge $e \in E$ can be either operational or in failed state, with respective probabilities p_e and $1 - p_e$. In this work, for simplicity we will assume that the capacity of the links is infinite, or alternatively, that the flow travelling through the links is negligible in comparison with their capacity. The rationale behind this assumption is to preclude the possibility of cascading failures, which may render our reliability analysis intractable. We will consider two cases:

- Univariate probabilistic model: the probability p_e is the same for all edges $e \in E$.
- Bivariate probabilistic model: there are two types of edges, those with a probability p of being operational ('highways'), and those with probability q of being operational ('secondary roads').

3. Univariate case

3.1 The reliability polynomial

For any subset $E' \subseteq E$, let $G' = (V, E')$ be the corresponding subgraph of G . Let us also define the binary function $\phi: G' \rightarrow \{0,1\}$ that returns 1 or 0 respectively if G' is operational or not. Under our probabilistic model, the subgraph G' will occur if its edges E' are operational, and the rest of edges $E - E'$ are not. This is a probabilistic equation that can be written as

$$P(G') = \prod_{e \in E'} p_e \prod_{e \notin E'} (1 - p_e) \quad (3.1)$$

Since reliability is the expected value of ϕ over all possible subgraphs of G , then we can define reliability as

$$\text{Rel}(G) = \sum_{G' \subseteq G} P(G') \phi(G') = \sum_{G' \text{ is a pathset}} P(G') \quad (3.2)$$

In our univariate model $p_e = p$, so finally the graph's reliability is

$$\text{Rel}(G, p) = \sum_{\substack{E' \subseteq E \\ G' \text{ is a pathset}}} p^{|E'|} (1-p)^{m-|E'|} \quad (3.3)$$

As we can clearly see, $\text{Rel}(G, p)$ is a polynomial in p , hence we may call it **the reliability polynomial of G** . This polynomial allows different forms that may simplify later calculations, for instance:

- let N_i be the number of pathsets with i edges. Hence the reliability polynomial can be expressed by its N -form:

$$\text{Rel}(G, p) = \sum_{i=0}^m N_i p^i (1-p)^{m-i} \quad (3.4)$$

- let C_i be the number of cutsets with i edges. Hence the reliability polynomial can be expressed by its C -form:

$$\text{Rel}(G, p) = 1 - \sum_{i=0}^m C_i (1-p)^i p^{m-i} \quad (3.5)$$

- the reliability polynomial has, finally, an expanded form:

$$\text{Rel}(G, p) = \sum_{k=0}^m \alpha_k p^k \quad (3.6)$$

The degree and the calculation of some coefficients of the expanded form are derived from the N -form: for any $i = 0, \dots, m$, any term $N_i p^i (1-p)^{m-i}$ in (3.4) is a polynomial of degree m such that

$$N_i p^i (1-p)^{m-i} = N_i p^i + \sum_{j=i+1}^{m-1} \beta_{i,j} p^j + N_i p^m \quad (3.7)$$

for some unknown real coefficients $\beta_{i,j}$. Thus, the reliability polynomial is a polynomial of degree m , and for any $k = 0, \dots, m-1$,

$$\alpha_k = N_k + \sum_{i < k} \beta_{i,k} \quad (3.8)$$

and

$$\alpha_m = \sum_{i=0}^m N_i \quad (3.9)$$

Some of the coefficients of the N -form of the reliability polynomial can be computed exactly. A well-known result from graph theory is that any spanning tree (a minpath) has $n-1$ edges, so

$$N_i = 0 \text{ if } i < n-1 \quad (3.10)$$

We can also easily compute N_{n-1} . Let $L = D - A$ be the Laplacian matrix of G , where D is the diagonal degree matrix, and A is the adjacency matrix of G . Now let $L_{i,j}$ denote the matrix obtained by deleting the i^{th} row and the j^{th} column from L . For any row i and column j , Kirchoff's Matrix Tree Theorem states that

$$N_{n-1} = (-1)^{i+j} \det(L_{i,j}) \quad (3.11)$$

Additionally, if c is the minimum cardinality of a cutset, then $C_i = 0$ for $i < c$. This means that

$$N_{m-i} = \binom{m}{m-i} \quad (3.12)$$

for $i < c$. The parameter c can be found by a network flow algorithm in polynomial time, repeated for every pair of vertices. However, the coefficients N_i for $n-1 < i < m-c$ remain unknown. Some of them may be computed efficiently, but the complexity of their calculation may render the analysis intractable -and out of scope in this study. Hence, not all of the above results regarding the exact computation of N -form coefficients will translate, in the expanded form (3.6), into exact results.

In conclusion, all we know about the expanded form of the reliability polynomial, from an exact computation perspective, is:

$$\left. \begin{aligned} &\bullet \alpha_i = 0 \text{ for all } i < n-1 \\ &\bullet \alpha_{n-1} = N_{n-1} \\ &\bullet \sum_{i=n-1}^m \alpha_i = 1 \end{aligned} \right\} \quad (3.13)$$

3.2 Polynomial interpolation of the reliability polynomial

For a given continuous function $f: [a, b] \in \mathbb{R} \rightarrow \mathbb{R}$ with known values in $n+1$ points x_0, \dots, x_n such that $a \leq x_0 < \dots < x_n \leq b$, polynomial interpolation consists in finding a polynomial $f_n \in P_n$, where P_n is the space of polynomials of degree at most $n \in \mathbb{N}$ with real coefficients, such that for all $i = 0, \dots, n$, $f_n(x_i) = f(x_i)$.

Going back to network science and our stochastic network with univariate edge failure, an interesting and obvious result from approximation theory states that a polynomial is its own polynomial interpolant.

Interpolation has involved long discussions regarding its potential convergence¹, as well as the best sets of interpolation points in order to avoid interpolation errors. Although theoretically there is no doubt about the interpolation's convergence when applied to a polynomial, since the result should be exact, we shall take into account the general result proven by Lloyd N. Trefethen [6] that states that the interpolation convergence is guaranteed² if Chebyshev points in $[a, b]$ are chosen as interpolation nodes. For $j = 0, \dots, n$, Chebyshev nodes are defined as

$$x_{n-j} = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{j\pi}{n}\right) \quad (3.14)$$

We normally apply interpolation techniques when f is a known function that is computationally too complex to evaluate at any given point, and when we can easily obtain the values $f(x_i)$ for a given set of interpolation points. In our study we are initially placed in a different position: $\text{Rel}(G, p)$ is not a known function, and *a priori* we do not have any information as to specific reliability values for any given p .

¹ Convergence in interpolation is defined as $\|f_n - f\|_\infty = \max\{|f_n(x) - f(x)|: x \in [a, b]\} \rightarrow 0$ when $n \rightarrow \infty$.

² As long as f is Lipschitz continuous, which in general terms means sufficiently smooth.

However, we can certainly rely on some of the previously obtained information, like (3.13). Also, we may assume that the reliability $\text{Rel}(G, p)$ for a given value of p can be approximated using simulation methods and powerful calculus software such as MATLAB (version R2019a for student use, © 1994-2020 The MathWorks, Inc.), and obtain the interpolation values with which we can apply this approximation method.

Below are described the steps followed in our proposed method:

Starting from a given graph $G = (V, E)$ and a given value p , let us perform a simulation where, for each edge $e \in E$, we will evaluate whether it remains operational or not: firstly generating a random value $r_e \in [0, 1]$, and then “turning off” the edge if $r_e > p$. After repeating this same operation for all edges, the resulting subgraph $G' \subseteq G$, and more specifically its operability $\phi(G')$, can be easily assessed using for instance the depth-first search (dfs) algorithm. Starting from an arbitrary node, this algorithm explores the graph in search of all the connected nodes, so only if n nodes are explored the subgraph is operational. After a sufficiently large number M of simulations, if G'_i is the subgraph resulting from the i^{th} simulation, then we can define the Simulated Reliability SR as

$$\text{SR}(G, p) = \sum_{i=1}^M \phi(G'_i) / M \approx \text{Rel}(G, p) \quad (3.15)$$

Let $0 \leq p_0 < \dots < p_k \leq 1$ be a set of probabilities, with $k \leq m$ (we do not want more than $m + 1$ interpolation points, as the reliability polynomial is of degree at most m). For each p_j , with $j = 0, \dots, k$, we know that

$$\text{Rel}(G, p_j) = \alpha_{n-1} p_j^{n-1} + \alpha_n p_j^n + \dots + \alpha_m p_j^m \quad (3.16)$$

We can therefore express this system of k equalities with matrices:

$$\begin{pmatrix} p_0^{n-1} & p_0^n & \dots & p_0^m \\ p_1^{n-1} & p_1^n & \dots & p_1^m \\ \vdots & \vdots & \ddots & \vdots \\ p_k^{n-1} & p_k^n & \dots & p_k^m \end{pmatrix} \begin{pmatrix} \alpha_{n-1} \\ \alpha_n \\ \vdots \\ \alpha_m \end{pmatrix} \approx \begin{pmatrix} \text{SR}(G, p_0) \\ \text{SR}(G, p_1) \\ \vdots \\ \text{SR}(G, p_k) \end{pmatrix} \quad (3.17)$$

Moreover, since we know the value $\alpha_{n-1} = N_{n-1}$, we can reduce the dimension of the unknown coefficients:

$$\begin{pmatrix} p_0^n & \dots & p_0^m \\ p_1^n & \dots & p_1^m \\ \vdots & \ddots & \vdots \\ p_k^n & \dots & p_k^m \end{pmatrix} \begin{pmatrix} \alpha_n \\ \vdots \\ \alpha_m \end{pmatrix} \approx \begin{pmatrix} \text{SR}(G, p_0) - \alpha_{n-1} p_0^{n-1} \\ \text{SR}(G, p_1) - \alpha_{n-1} p_1^{n-1} \\ \vdots \\ \text{SR}(G, p_k) - \alpha_{n-1} p_k^{n-1} \end{pmatrix} \quad (3.18)$$

Above we have a system of $k + 1$ equations and $m - n + 1$ unknowns, so let us define

$$k = m - n \quad (3.19)$$

Now the first matrix is a square matrix, which determinant is

$$\det \begin{pmatrix} p_0^n & \dots & p_0^m \\ p_1^n & \dots & p_1^m \\ \vdots & \ddots & \vdots \\ p_{m-n}^n & \dots & p_{m-n}^m \end{pmatrix} = (p_0^n \dots p_{m-n}^n) \det \begin{pmatrix} 1 & p_0 & \dots & p_0^{m-n} \\ 1 & p_1 & \dots & p_1^{m-n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & p_{m-n} & \dots & p_{m-n}^{m-n} \end{pmatrix} \quad (3.20)$$

We can easily identify the right-term above as a Vandermonde determinant, which is non-zero when $p_v \neq p_v$ if $v \neq v$, and when none of the probabilities in our set is zero, which is the case if we choose the interpolation points adequately.

We have proven that a solution exists, and is unique, to the computation of the univariate reliability polynomial in its expanded form, using the below formula

$$\begin{pmatrix} \alpha_n \\ \vdots \\ \alpha_m \end{pmatrix} \approx \begin{pmatrix} p_0^n & \cdots & p_0^m \\ p_1^n & \cdots & p_1^m \\ \vdots & \ddots & \vdots \\ p_{m-n}^n & \cdots & p_{m-n}^m \end{pmatrix}^{-1} \begin{pmatrix} \text{SR}(G, p_0) - \alpha_{n-1} p_0^{n-1} \\ \text{SR}(G, p_1) - \alpha_{n-1} p_1^{n-1} \\ \vdots \\ \text{SR}(G, p_{m-n}) - \alpha_{n-1} p_{m-n}^{n-1} \end{pmatrix} \quad (3.21)$$

and if the $m - n + 1$ interpolation probabilities are chosen as the set of $m - n + 2$ Chebyshev points in $[0,1]$ where the value 0 is removed.

As a recapitulation, our proposed method to approximate the reliability polynomial of a stochastic network with univariate edge failure under the all-terminal reliability condition combines results from graph theory, polynomial interpolation at Chebyshev points, and simulation. The following diagram summarizes the main steps of this method.

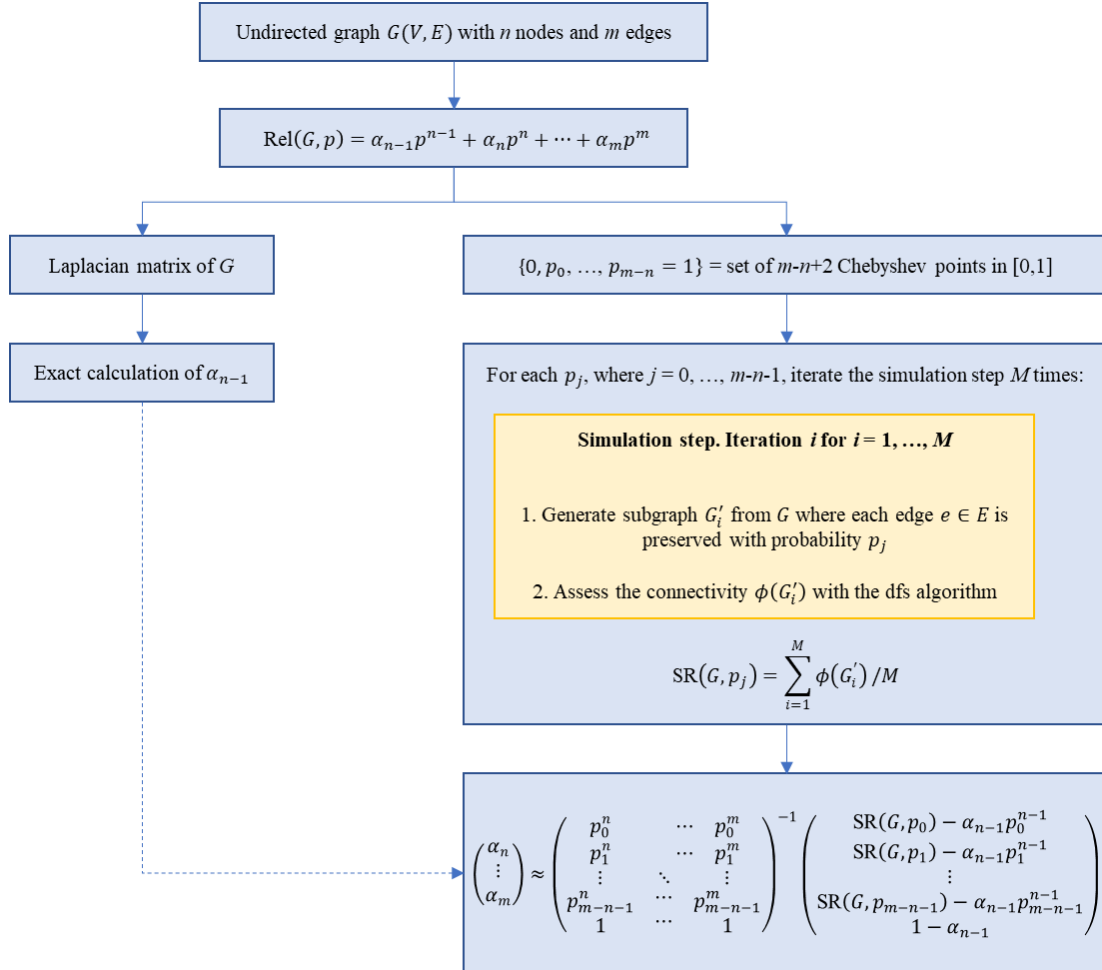


Figure 2: Diagram of the proposed method for the univariate case

3.3 Practical application

For the reasons exposed above, there are only few examples of stochastic networks with univariate edge failure where the reliability polynomial is known, and these instances model very simple systems. Nevertheless, one of these systems can be used to test the simulation-interpolation method proposed.

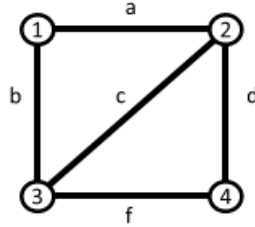


Figure 3: The graph $K_4 - \{e\}$

Let us consider the graph in figure 3. For all-terminal reliability, it is fairly simple to list all the pathsets with 3, 4 and 5 edges: $\{acd, acf, abd, abf, adf, bcd, bcf, bdf, abcd, abcf, abdf, acdf, bcdf, abcdf\}$, and obtain the following N -form of the reliability polynomial

$$\text{Rel}(G, p) = p^5 + 5p^4(1 - p) + 8p^3(1 - p)^2$$

from where we get the expanded form (which we will compare our results against)

$$\text{Rel}(G, p) = 4p^5 - 11p^4 + 8p^3$$

Following the formula (3.11) we can exactly compute $\alpha_3 = N_3$:

$$\begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix} \rightarrow \begin{vmatrix} -1 & -1 & 0 \\ 3 & -1 & -1 \\ -1 & 3 & -1 \end{vmatrix} = -8$$

Starting from the Laplacian matrix of the graph, and then removing the 4th row and the 1st column, we obtain $\alpha_3 = (-1)^5(-8) = 8$.

Secondly, we want to approximate the coefficients α_4 and α_5 using the proposed method: the interpolation points in $[0,1]$ will be defined as the non-zero points in the set of three Chebyshev points in $[0,1]$. Hence, our interpolation points will be $p_0 = 0.5$ and $p_1 = 1$. In this extremely simple case we only need to run the simulation step for p_0 , since the exact value of the reliability polynomial is necessarily 1 for $p = p_1$. The appropriate code has been written in MATLAB, and the simulation step has been set to execute $M = 10^5$ iterations. The obtained value for the simulated reliability is

$$\text{SR}(G, p_0) = 0.4365$$

With the same MATLAB code, we have implemented the simple matrix formula (3.21), obtaining the final result

$$\alpha_4 \approx -11.0317 \text{ and } \alpha_5 \approx 4.0317$$

This result is very close to the exact solution, and although there is definitely an absolute error $\varepsilon = 0.0317$ in both values, we shall apply a rounding procedure, knowing that the coefficients should be integers, which would result in the exact solution.

Now let us take a more complex example: the simplified representation of the island of Ibiza's road map, as shown in figure 1. This time, the MATLAB code also includes the computation of α_{n-1} . The results, using the same method as in the first example, are summarized in the below table:

Coefficient	Approximation
α_{23}	12
α_{24}	1.4460×10^4
α_{25}	-9.7547×10^5
α_{26}	1.8175×10^7
α_{27}	-1.4874×10^8
α_{28}	6.4159×10^8
α_{29}	-1.5921×10^9
α_{30}	2.3548×10^9
α_{31}	-2.0524×10^9
α_{32}	9.7363×10^8
α_{33}	-1.9403×10^8

3.4 Quality assessment of the proposed method and calibration

The quality of the proposed method can be assessed to the extent that the reliability polynomial is known only in rare occasions, with extremely simple graphs.

With the graph $K_4 - \{e\}$, like with any graph with known reliability polynomial, the quality assessment is immediate and can be measured in terms of the relative difference (RD) of the obtained polynomial (with approximated coefficients) at $p = p_r$, for a uniformly random value $p_r \in [0,1]$, with respect to the exact graph's reliability $\text{Rel}(G, p_r)$.

As part of the method's quality assessment, the number of iterations at the simulation step can act as a calibration knob between error and computational cost. Hence, the quality assessment is performed for different values of M , and the results show, in addition to the RD, the time elapsed at the simulation step.

It is also important to point out that the MATLAB code written to perform our method (see Annexes) makes use of the parallelization capacity of the software. The simulation step is designed to use multiple threads in parallel, taking advantage of the various cores of the personal computer's CPU used in this study (with a processor Intel® Core™ i7-8700 CPU @ 3.20GHz).

Finally, in order to mitigate the effect of potential outliers, for each value M , the experiment is repeated with 10 different random values p_r and the results are the averages.

	Number of simulations per interpolation point					
	10	10 ²	10 ³	10 ⁴	10 ⁵	10 ⁶
Time (seconds)	3,11E-02	7,07E-02	1,73E-01	1,34E+00	1,15E+01	1,10E+02
RD	2,16E-01	5,97E-02	1,76E-02	7,39E-03	2,00E-03	3,65E-04

Figure 4: Numerical performance of the proposed method for the univariate case

The above results show that while the time spent on each experiment grows logically at the same rate as the number of simulations increase, the relative difference between the graph's reliabilities (calculated with the exact reliability polynomial and the approximated reliability polynomial) decreases at an inversely proportional rate. The graphical analysis will enable further interpretation of these results:

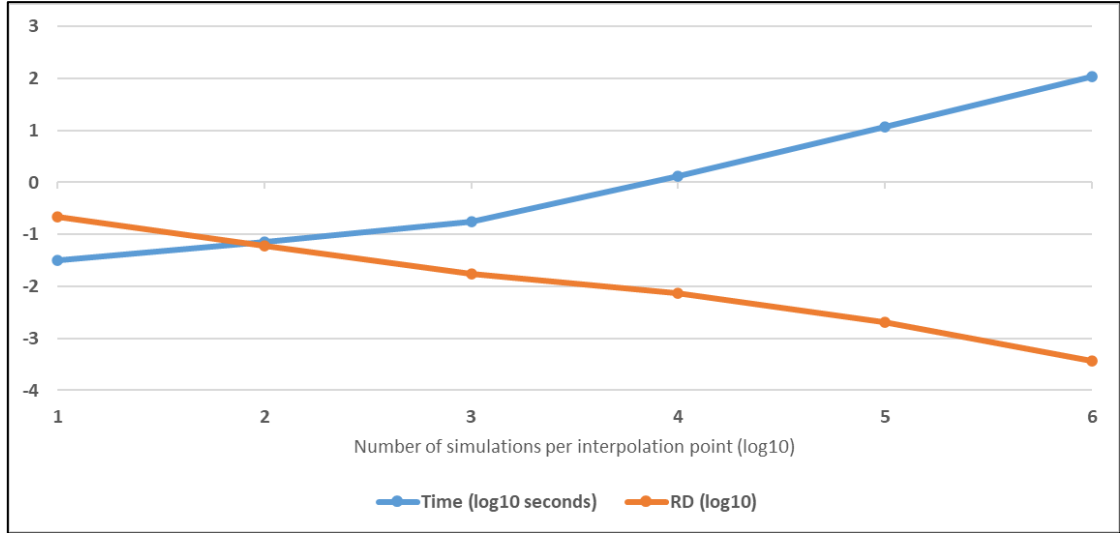


Figure 5: Graphical performance of the proposed method for the univariate case

The logarithmic improvement of the RD as an increasing number of simulations is used to obtain the simulated reliability at the interpolation points seems to be linear. The RD reduction factor when the number of simulations is multiplied by 10 is in average $2.9 \cdot 10^{-1}$ (with a standard deviation of $0.8 \cdot 10^{-1}$). Furthermore, from 10^5 to 10^6 simulations, the reduction factor reached its minimum, $1.8 \cdot 10^{-1}$.

At the highest number of simulations ($M = 10^6$), the RD between the approximation and the exact reliability is considerably low, while each experiment takes in average nearly 2 minutes, which, considering the fact that only one interpolation point is driving most calculations, is a significant amount of time.

4. Bivariate case

The assumption that all edges have the same reliability is too restrictive. In practice, each edge may have its own probability of failing, or at least, there is a set of failure probabilities to choose from. The second simplest case is when we have two types of edges, with probabilities p and q of being operational. This leads to a bivariate reliability polynomial in the indeterminates p and q .

This model is suitable for several real-life situations. For example, in a road network there are basically two types of roads: highways and conventional roads. Highways are usually more straightforward, with less curves, and they have several lanes for each direction, enabling different speeds. They may also have a physical division that separates both directions, protection against crossing animals, better signaling, etc. All this leads to a smaller probability of accidents, thus a greater reliability.

4.1 The reliability polynomial

Following the same reasoning and terminology as in the univariate case, the reliability of a graph G is given by (3.2), where again $\phi(G') = 1$ if G' is operational under all-terminal reliability, $\phi(G') = 0$ otherwise, and $P(G')$ is the occurrence probability of the subgraph G' .

Let $E = E_p \cup E_q$, where the operational probability is p for the edges of E_p (we may call them p -edges), and q for the edges of E_q (q -edges). Let us define $m_p = |E_p|$ and $m_q = |E_q|$, with $|E| = m = m_p + m_q$. We may assume, without loss of generality, that $m_p \geq m_q$.

Thus, if $G' = (V, E')$ and $E' = E'_p \cup E'_q$, the reliability of G is

$$\text{Rel}(G, p, q) = \sum_{\substack{E' \subseteq E \\ G' \text{ is a pathset}}} p^{|E'_p|} (1-p)^{m_p - |E'_p|} q^{|E'_q|} (1-q)^{m_q - |E'_q|} \quad (4.1)$$

The function $\text{Rel}(G, p, q)$ is clearly a bivariate polynomial in p and q , **the reliability polynomial of G** , and like in the univariate case, it allows multiple forms, for instance:

- if $N_{i,j}$ is the number of pathsets with i p -edges and j q -edges, then its N -form is

$$\text{Rel}(G, p, q) = \sum_{i=0}^{m_p} \sum_{j=0}^{m_q} N_{i,j} p^i q^j (1-p)^{m_p-i} (1-q)^{m_q-j} \quad (4.2)$$

- similarly, if $C_{i,j}$ is the number of cutsets with i p -edges and j q -edges, then its C -form is

$$\text{Rel}(G, p, q) = 1 - \sum_{i=0}^{m_p} \sum_{j=0}^{m_q} C_{i,j} (1-p)^i (1-q)^j p^{m_p-i} q^{m_q-j} \quad (4.3)$$

- the expanded form of the reliability polynomial of G is

$$\text{Rel}(G, p, q) = \sum_{k_1=0}^{m_p} \sum_{k_2=0}^{m_q} \alpha_{k_1, k_2} p^{k_1} q^{k_2} \quad (4.4)$$

For all of the above forms, the exact computation of all coefficients is intractable, but just like in the univariate case, we can extract some exact information about these coefficients. This partial understanding of the reliability polynomial will help refine the simulation-interpolation approach to its approximation in the next section.

The univariate case has showed us that a minpath has $n - 1$ edges, so

$$N_{i,j} = 0 \text{ if } i + j < n - 1 \quad (4.5)$$

and if N_{n-1} is the number of minpaths of G , which we can easily compute with the formula (3.11), then

$$\sum_{i+j=n-1} N_{i,j} = N_{n-1} \quad (4.6)$$

These last results can significantly reduce the length of the expanded form. Indeed, in the N -form, for any $i \leq m_p$ and any $j \leq m_q$, each of the addends $N_{i,j} p^i q^j (1-p)^{m_p-i} (1-q)^{m_q-j}$ is a polynomial in the indeterminates p and q with the term of maximal total degree equal to $N_{i,j} p^{m_p} q^{m_q}$ and the term of minimal total degree equal to $N_{i,j} p^i q^j$. Thus, we can deduce that

$$(4.7)$$

$$\text{Rel}(G, p, q) = \sum_{\substack{k_1=m_p \\ k_2=m_q \\ k_1+k_2=n-1}} \alpha_{k_1, k_2} p^{k_1} q^{k_2}$$

with

$$\sum_{k_1+k_2=n-1} \alpha_{k_1, k_2} = N_{n-1} \quad (4.8)$$

and

$$\sum_{\substack{k_1=m_p \\ k_2=m_q \\ k_1+k_2=n-1}} \alpha_{k_1, k_2} = 1 \quad (4.9)$$

The univariate case also evidences the fact that, although we may be able to compute the coefficients N_{m-i} for $i < c$, where c is the minimum cardinality of a cutset, each non-zero coefficient N_μ for any $\mu \leq m$ will contribute to the coefficients of the expanded form α_k for all $\mu \leq k \leq m$. Thus, the exact computation of more coefficients (or sums of groups of coefficients) in the expanded form, other than (4.8) and (4.9), is intractable and out of scope in this study.

We can group the terms with the same total degree to rewrite the extended form of the reliability polynomial as the sum

$$\text{Rel}(G, p, q) = \sum_{k_1+k_2=n-1} \alpha_{k_1, k_2} p^{k_1} q^{k_2} + \dots + \sum_{k_1+k_2=m} \alpha_{k_1, k_2} p^{k_1} q^{k_2} \quad (4.10)$$

where each of the addends can be rewritten using only one exponent:

$$\begin{aligned} \sum_{k_1+k_2=K} \alpha_{k_1, k_2} p^{k_1} q^{k_2} &= \sum_{\substack{k_1=\max\{0; K-m_q\} \\ \min\{m_p; K\}}}^{\min\{m_p; K\}} \alpha_{k_1, K-k_1} p^{k_1} q^{K-k_1} \\ &= \sum_{k_2=\max\{0; K-m_p\}} \alpha_{K-k_2, k_2} p^{K-k_2} q^{k_2} \end{aligned} \quad (4.11)$$

for $n-1 \leq K \leq m$.

Finally, we can group the terms with the same partial degree in the same indeterminate:

$$\text{Rel}(G, p, q) = \sum_{k_1=\max\{0; n-1-m_q\}}^{m_p} p^{k_1} \left[\sum_{k_2=\max\{0; n-1-k_1\}}^{m_q} \alpha_{k_1, k_2} q^{k_2} \right] \quad (4.12)$$

or equivalently

$$\text{Rel}(G, p, q) = \sum_{k_2=\max\{0; n-1-m_p\}}^{m_q} q^{k_2} \left[\sum_{k_1=\max\{0; n-1-k_2\}}^{m_p} \alpha_{k_1, k_2} p^{k_1} \right] \quad (4.13)$$

4.2 Polynomial interpolation of the bivariate reliability polynomial

Several techniques of multivariate polynomial interpolation have been developed through relatively recent research studies. Some of the most known of these techniques are the use of a multivariate Vandermonde matrix, the Hermite interpolation, or the Newton interpolation (which we will explore later on). While the univariate polynomial interpolation guarantees the uniqueness and existence of the solution (as long as few requirements are met), the multivariate case is more complex. Determining whether a multivariate interpolation has a unique solution (*poisedness*) in the space of polynomials with given total and/or partial degrees is difficult. This matter requires a thorough study of the interpolation nodes, the polynomial space, the specific interpolation technique. Furthermore, the study of multivariate interpolation techniques is a flourishing field of study, yet filled with unresolved questions.

With this in mind, and given the fact that the formulas (4.12) and (4.13) suggest the possibility of applying a univariate polynomial interpolation technique to the approximation of the bivariate reliability polynomial, we are proposing a first method to approximate the reliability polynomial with a univariate approach.

4.2.1 Bidimensional univariate polynomial interpolation (BUPI)

Let us take the equation (4.12) as the starting point, with p as the indeterminate of study and a fixed value $Q \in [0,1]$ of q . The reliability polynomial becomes a univariate polynomial in p , of degree m_p , and whose coefficients are zero for all terms of degree lower than $\max\{0; n - 1 - m_q\}$:

$$\text{Rel}(G, p, Q) = \sum_{k_1=\max\{0; n-1-m_q\}}^{m_p} A_{k_1} p^{k_1} \quad (4.14)$$

We know that we can approximate the unknown coefficients A_{k_1} through polynomial interpolation in a set of $m_p - \max\{0; n - 1 - m_q\} + 1$ points, and that these points can guarantee an optimal result (in terms of minimized error) if they are Chebyshev points in $[0,1]$.

Similarly, if we consider the equation (4.13) with q as the indeterminate and a fixed value $P \in [0,1]$ of p , then we obtain the univariate polynomial:

$$\text{Rel}(G, P, q) = \sum_{k_2=\max\{0; n-1-m_p\}}^{m_q} B_{k_2} q^{k_2} \quad (4.15)$$

that we can approximate using $m_q - \max\{0; n - 1 - m_p\} + 1$ Chebyshev points in $[0,1]$.

Again, for the fixed value Q and each interpolation point p_γ (resp. P and q_δ), we will approximate the value $\text{Rel}(G, p_\gamma, Q)$, resp. $\text{Rel}(G, P, q_\delta)$, with the simulated reliability by applying the same simulation procedure, with M iterations, as in the univariate case.

Once the coefficients A_{k_1} and B_{k_2} have been approximated via univariate polynomial interpolation, the coefficients of the expanded form of $\text{Rel}(G, P, q)$ can be obtained following the below scheme:

1. The bivariate reliability polynomial allows an expanded form, with coefficients that can be stored in a matrix (assuming that $m_p \geq m_q$):

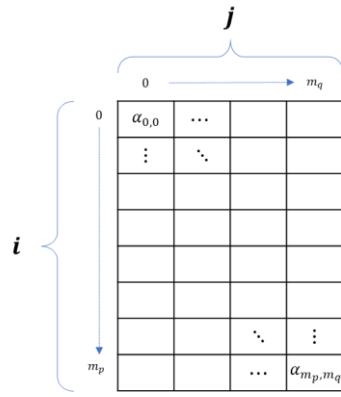


Figure 6: Matrix of the coefficients of the expanded form

2. These coefficients are zero if $i + j < n - 1$, and their sum when $i + j = n - 1$ is equal to N_{n-1} :

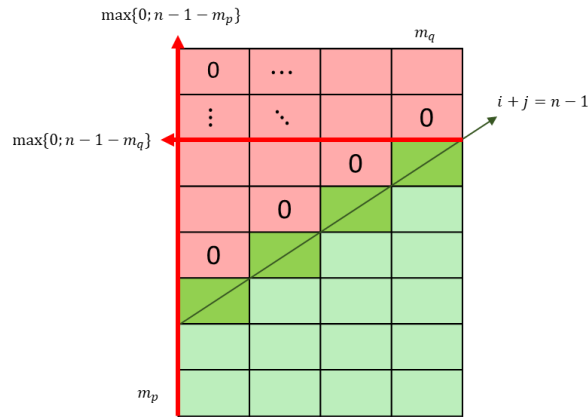


Figure 7: Minimal coefficient matrix

3. Each of the non-zero coefficients is also a coefficient in two univariate polynomials, one for each indeterminate p and q :

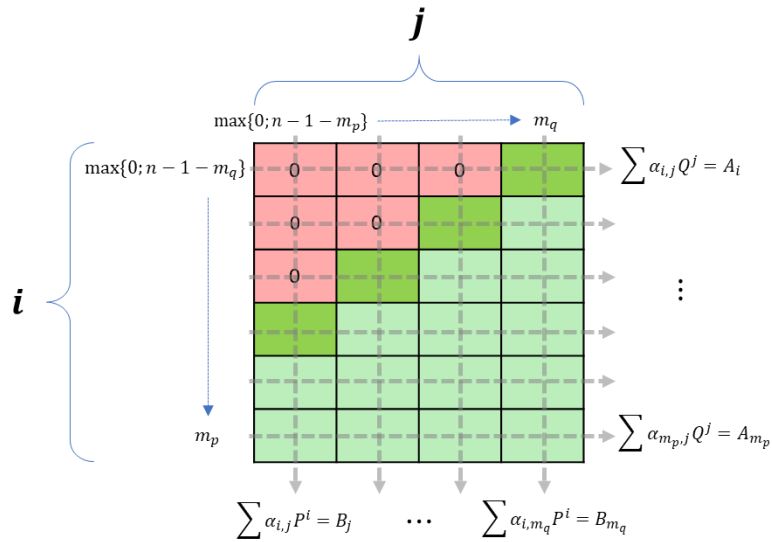


Figure 8: Two sets of equations after a bidimensional univariate interpolation

Finally, this method provides the above system of $[m_q - \max\{0; n - 1 - m_p\} + 1] + [m_p - \max\{0; n - 1 - m_q\} + 1]$ equations, where each unknown $\alpha_{i,j}$ appears in two and only two equations with different known coefficients (P^i and Q^j). Using MATLAB's capabilities of matrix calculation, this system is the basic ingredient to approximate the reliability polynomial. Additionally, in order to prevent singularity or bad scaling, the system is completed with two extra equations, (4.8) and (4.9), which add a component of exactness.

As a recapitulation, the following diagram shows the different steps of this method:

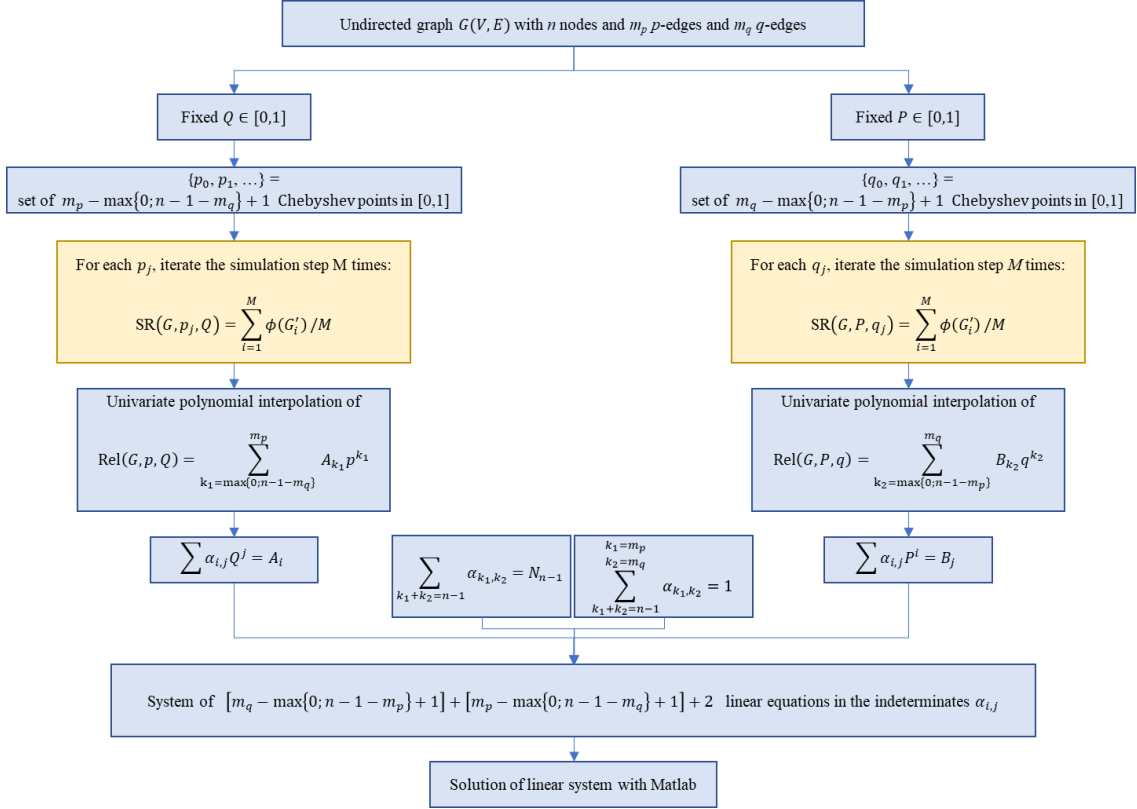


Figure 9: Diagram of the proposed method: BUPI

4.2.2 Newton's bivariate polynomial interpolation (NBPI) in a rectangular basis

The second method we are proposing to approximate the reliability polynomial in the bivariate case is a variant of Newton's bivariate polynomial interpolation. Unlike univariate polynomial interpolation, this technique cannot guarantee the uniqueness of the solution, nor does it enable setting all the additional conditions to the resulting polynomial interpolant as we did with the univariate method (such as only dealing with a specific number of non-zero coefficients, or ensuring that the sum of some specific coefficients adds up to N_{n-1}). This method is only based on the value of the graph's reliability in a set of different bivariate points $(p, q) \in [0,1]^2$, and on recursively searching for the best approximation in a sequence of spaces of bivariate polynomials in the indeterminates p and q with increasing partial degrees (the maximum exponents of p and q) and total degree (the maximum sum of the exponents of p and q in the same addend).

The poisedness of this method depends, to a large extent, on the choice of the interpolation nodes (p, q) in $[0,1]^2$, and the application of this technique is relatively easy using a recursive algorithm,

explained hereafter: let us assume that we aim to approximate a function $f(x, y)$ with a polynomial $p(x, y)$ which has known partial degrees d_1 and d_2 (with $d_1 > d_2$), and known total degree $d = d_1 + d_2$, so

$$p(x, y) = \sum_{i=0}^{d_1} \sum_{j=0}^{d_2} a_{i,j} x^i y^j \quad (4.16)$$

with $a_{d_1, d_2} \neq 0$. The optimal set of interpolation nodes is a rectangular basis of $(d_1 + 1)(d_2 + 1)$ points

$$S = \{(x_i, y_j) \in [0, 1]^2 \subset \mathbb{R}^2 : 0 \leq i \leq d_1, 0 \leq j \leq d_2\} \quad (4.17)$$

such that $x_0 < \dots < x_{d_1}$ and $y_0 < \dots < y_{d_2}$, and the values $f(x_i, y_j)$ are known. With these conditions, the polynomial $p(x, y)$ is unique if

$$p(x, y) = X^t \cdot P \cdot Y \quad (4.18)$$

where:

$$\left. \begin{aligned} & \bullet \quad X = \begin{pmatrix} 1 \\ x - x_0 \\ (x - x_0)(x - x_1) \\ \vdots \\ (x - x_0)(x - x_1) \dots (x - x_{d_1-1}) \end{pmatrix}, \quad Y = \begin{pmatrix} 1 \\ y - y_0 \\ (y - y_0)(y - y_1) \\ \vdots \\ (y - y_0)(y - y_1) \dots (y - y_{d_2-1}) \end{pmatrix} \\ & \bullet \quad P = \begin{pmatrix} p_{0,0}^{(0)} & p_{0,1}^{(1)} & p_{0,2}^{(2)} & \dots & p_{0,d_2}^{(d_2)} \\ p_{1,0}^{(1)} & p_{1,1}^{(1)} & p_{1,2}^{(2)} & \dots & p_{1,d_2}^{(d_2)} \\ p_{2,0}^{(2)} & p_{2,1}^{(2)} & p_{2,2}^{(2)} & \dots & p_{2,d_2}^{(d_2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{d_2,0}^{(d_2)} & p_{d_2,1}^{(d_2)} & p_{d_2,2}^{(d_2)} & \dots & p_{d_2,d_2}^{(d_2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{d_1,0}^{(d_1)} & p_{d_1,1}^{(d_1)} & p_{d_1,2}^{(d_1)} & \dots & p_{d_1,d_2}^{(d_1)} \end{pmatrix} \\ & \bullet \quad p_{i,j}^{(0)} = f(x_i, y_j) \\ & \bullet \quad p_{i,j}^{(k)} = \begin{cases} \frac{p_{i,j}^{(k-1)} - p_{i-1,j}^{(k-1)}}{x_i - x_{i-k}} & \text{if } i \geq k \text{ and } j < k \\ \frac{p_{i,j}^{(k-1)} - p_{i,j-1}^{(k-1)}}{y_j - y_{j-k}} & \text{if } i < k \text{ and } j \geq k \\ \frac{p_{i,j}^{(k-1)} + p_{i-1,j-1}^{(k-1)} - p_{i-1,j}^{(k-1)} - p_{i,j-1}^{(k-1)}}{(x_i - x_{i-k})(y_j - y_{j-k})} & \text{if } i \geq k \text{ and } j \geq k \\ p_{i,j}^{(k-1)} & \text{if } i < k \text{ and } j < k \end{cases} \end{aligned} \right\} \quad (4.19)$$

In this study, the function f that will be approximated using this method is a graph's reliability, but with a slight nuance that will force the final interpolant to have its coefficients with the lowest indices equal to zero. As per (4.12) and (4.13), the minimal exponent of p in $\text{Rel}(G, p, q)$ is $\max\{0; n - 1 - m_q\}$.

Likewise, the minimal exponent of q in $\text{Rel}(G, p, q)$ is $\max\{0; n - 1 - m_p\}$. Hence the interpolated function will be

$$f(p, q) = p^{-\max\{0; n-1-m_q\}} \cdot q^{-\max\{0; n-1-m_p\}} \cdot \text{SR}(G, p, q) \quad (4.20)$$

where, once again, the value of the graph's reliability is replaced with its simulated value. The polynomial interpolant obtained with this method will have partial degrees $d_1 = m_p - \max\{0; n - 1 - m_q\}$ and $d_2 = m_q - \max\{0; n - 1 - m_p\}$, and total degree $d = d_1 + d_2$.

Finally, some additional remarks about the resulting approximation of the reliability polynomial using the Newton bivariate polynomial interpolation that must be taken into account are:

- the connection between the coefficients $a_{i,j}$ of the resulting interpolant and the coefficients of the graph's reliability polynomial is $a_{i,j} \approx \alpha_{i+\max\{0; n-1-m_q\}, j+\max\{0; n-1-m_p\}}$.
- there will still be some coefficients with low indices, where $i + j < n - 1$, that will likely be non-zero in the result.
- the equations (4.8) and (4.9) cannot be taken into account with this method.

As a recapitulation, the below diagram shows the different steps of this method:

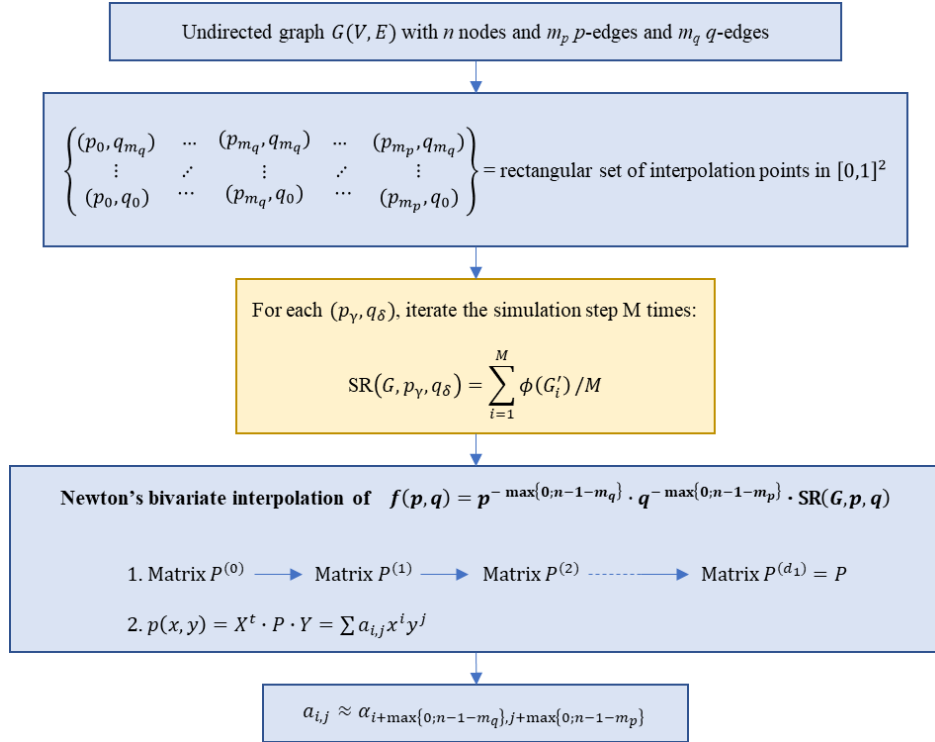


Figure 10: Diagram of the proposed method: NBPI

4.3 Practical application

Similarly to the univariate case, the proposed simulation-interpolation based methods have been put into practice with the same two different graphs, but with the bivariate component.

Also, in order to apply both methods in the most efficient manner, the sets of interpolation nodes of each method have been merged. The BUPI method requires two sets of points (p_i, Q) and (P, q_j) which should be two sets of Chebyshev points in order to optimize the method (for this study P and Q will be

chosen as centered in $[0,1]$ as possible). Therefore, the NBPI method will expand these sets of points into a rectangular basis, as shown in the following figure. In this manner, the set of simulation runs for each point of this rectangular basis will provide both methods with their simulated reliabilities.

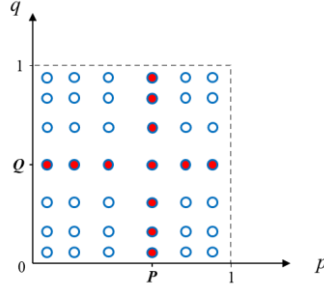


Figure 11: Example scheme of interpolation points for the BUPI (filled in red) and the NBPI (all)

The first graph is $K_4 - \{e\}$, where three edges have probability p of remaining operational, whereas this probability is q for the remaining two edges.

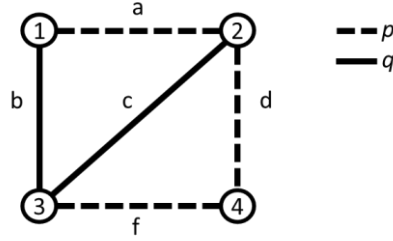


Figure 12: Bivariate $K_4 - \{e\}$

It is relatively simple to develop the N-form of $\text{Rel}(G, p, q)$, and obtain its expanded form:

$$\text{Rel}(G, p, q) = 4p^3q^2 - 5p^3q + p^3 - 6p^2q^2 + 5p^2q + 2pq^2$$

Using the first of the proposed methods, the BUPI, with the number of iterations in each simulation step set at $M = 10^5$, the result is

$$\text{Rel}(G, p, q) \approx (3.98)p^3q^2 - (5.02)p^3q + (0.99)p^3 - (5.96)p^2q^2 + (5.04)p^2q + (1.96)pq^2$$

This highest absolute error of the coefficients is $\varepsilon = 0.0448$ although if we had chosen to round our results (since we know the coefficients are integers), then we would have obtained the exact coefficients.

With the second of the proposed methods, the NBPI, the coefficients corresponding to the terms in p , p^2 and pq , which should be zero are non-zero, although their values are negligible (respectively, -6.64×10^{-3} , -1.26×10^{-2} and -2.96×10^{-2}). If we ignore these, the rest of the resulting polynomial is:

$$\text{Rel}(G, p, q) \approx (4.12)p^3q^2 - (5.16)p^3q + (1.02)p^3 - (6.14)p^2q^2 + (5.19)p^2q + (2.02)pq^2$$

The highest absolute error in these coefficients is $\varepsilon = 0.1649$, although the non-zero coefficients that we ignored may actually help adjust this error with respect to the known exact reliability polynomial.

The second graph is the road map of Ibiza, where this time we have taken into account the road type, and assigned each road or edge to one type: highways, with operability probability q , and secondary roads where this probability is p .

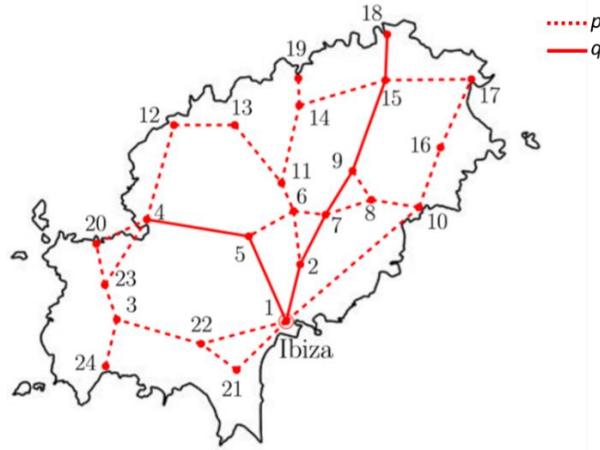


Figure 13: A simplified road map of Ibiza (with two types of roads)

Now we are faced with a totally different situation: we have no knowledge as to this graph's reliability polynomial.

However, using the BUPI method with $M = 10^5$, we obtain the following coefficients (in scientific format):

$\alpha_{i,j}$	$j=0$	1	2	3	4	5	6	7
$i = 16$	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	2,72E+02
17	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	4,12E+04	-1,05E+05
18	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	-1,75E+05	6,22E+05	0,00E+00
19	0,00E+00	0,00E+00	0,00E+00	0,00E+00	2,07E+04	-1,58E+06	0,00E+00	0,00E+00
20	0,00E+00	0,00E+00	0,00E+00	1,09E+06	2,63E+06	0,00E+00	0,00E+00	0,00E+00
21	0,00E+00	0,00E+00	-2,52E+06	-3,95E+06	0,00E+00	0,00E+00	0,00E+00	0,00E+00
22	0,00E+00	2,57E+06	5,20E+06	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
23	-1,37E+06	-4,63E+06	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
24	3,16E+06	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
25	-1,48E+06	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
26	2,95E+05	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00

One observation we can make is that the sum of coefficients where $i + j = n - 1$ does not equal N_{n-1} . This is probably due to the fact that in the system of equations that this method builds, the one corresponding to $N_{n-1} = 12$ has little influence on the solution as opposed to the equations corresponding to the values A_{k_1} and B_{k_2} . For instance, $A_{26} = 4.2 \cdot 10^6$.

The situation for the NBPI method is the same: there is no exact set of coefficients that we can compare our results with. However in this case the coefficient that we know are zero (light red upper left triangle of our matrix) have important absolute vales.

$\alpha_{i,j}$	$j = 0$	1	2	3	4	5	6	7
$i = 16$	7,50E+03	-3,14E+05	3,73E+06	-1,92E+07	4,99E+07	-6,87E+07	4,76E+07	-1,31E+07
17	-5,90E+05	2,47E+07	-2,93E+08	1,51E+09	-3,93E+09	5,40E+09	-3,75E+09	1,03E+09
18	1,33E+07	-5,55E+08	6,59E+09	-3,39E+10	8,82E+10	-1,21E+11	8,42E+10	-2,32E+10
19	-1,33E+08	5,55E+09	-6,59E+10	3,39E+11	-8,84E+11	1,22E+12	-8,43E+11	2,32E+11
20	7,13E+08	-2,98E+10	3,54E+11	-1,82E+12	4,74E+12	-6,52E+12	4,53E+12	-1,25E+12
21	-2,25E+09	9,43E+10	-1,12E+12	5,77E+12	-1,50E+13	2,07E+13	-1,43E+13	3,94E+12
22	4,41E+09	-1,85E+11	2,19E+12	-1,13E+13	2,94E+13	-4,04E+13	2,81E+13	-7,72E+12
23	-5,42E+09	2,27E+11	-2,69E+12	1,39E+13	-3,61E+13	4,96E+13	-3,44E+13	9,48E+12
24	4,06E+09	-1,70E+11	2,02E+12	-1,04E+13	2,70E+13	-3,72E+13	2,58E+13	-7,10E+12
25	-1,70E+09	7,11E+10	-8,44E+11	4,35E+12	-1,13E+13	1,56E+13	-1,08E+13	2,97E+12
26	3,05E+08	-1,27E+10	1,51E+11	-7,80E+11	2,03E+12	-2,79E+12	1,94E+12	-5,34E+11

Also, all these coefficients are completely different from those obtained in the previous experiment with the BUPI.

4.4 Quality assessment of the proposed methods and calibration

Like in the univariate case, we want to evaluate our proposed methods by checking the value of the approximated polynomial against the reliability of the graph in a random point in $[0,1]^2$. This real reliability is only known in the case of the graph $K_4 - \{e\}$, so the quality assessment of the proposed methods will be based on these results only.

This error assessment will be tested applying different values of M to the number of simulations run in each experiment (to obtain the reliability values at the interpolation points). Also, for each value M and each experiment, we will run 10 different tests (with 10 different random reference values (p_r, q_r)) and calculate the average relative error in order to mitigate the effect of potential outliers.

In parallel to measuring the error, we will also keep track of the time needed to perform each experiment, and show the mean duration of the test.

The relevant code has been written in MATLAB (see Annexes).

		Number of simulations per interpolation point					
		10	10 ²	10 ³	10 ⁴	10 ⁵	10 ⁶
BUPI	Time (seconds)	1,79E+00	3,96E-01	1,07E+00	7,19E+00	5,74E+01	5,50E+02
	RD	3,24E-01	1,57E-01	5,30E-02	9,19E-03	2,94E-03	1,32E-03
NBPI	Time (seconds)	3,38E+00	7,49E-01	1,96E+00	1,30E+01	1,03E+02	9,90E+02
	RD	4,36E-01	8,13E-02	3,02E-02	6,25E-03	3,95E-03	1,67E-03

Figure 14: Numerical performance of the proposed methods for the bivariate case

Again, there is a clear improvement in the RD obtained with both methods as a larger number of simulations is considered, while the time spent in calculations increases. Another logical result is the fact that, in average, the time spent on the NBPI method (which uses 3x3 interpolation points in this case) is 1.8 times greater than for the BUPI method (which uses only 3+3-1 of those points).

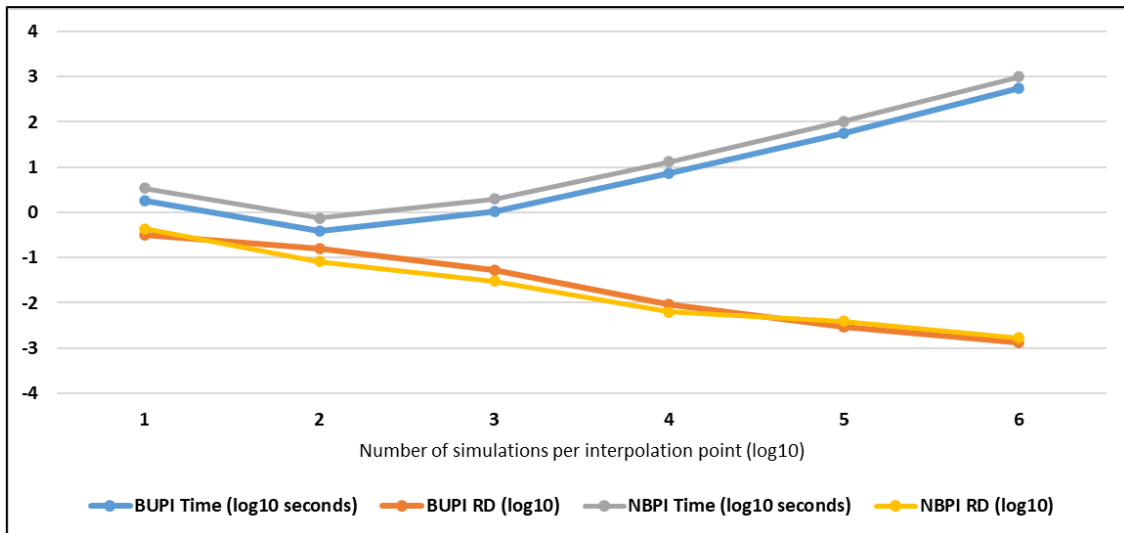


Figure 15: Graphical performance of the proposed methods for the bivariate case

In terms of logarithmic RD improvement, both methods show a linear behavior as the number of simulations increase. In fact, their performances seem very similar, even without the logarithmic scale adjustment. It is the performance in terms of time that appears to make the difference, and the BUPI method achieves better results in this regard. In fact, this method has also provided slightly better results in terms of RD for 10^5 and 10^6 simulations.

Looking more closely at the reduction of the RD with the BUPI method as the number of simulations increases by 10, the average reduction factor is $3.6 \cdot 10^{-1}$ (with a standard deviation of $1.8 \cdot 10^{-1}$). As opposed to the univariate case, the reduction factor is not as predictable although in average it is highly performant.

Lastly, it is worth noting that for 10^6 simulations, each experiment with the BUPI method took in average more than 9 minutes to be completed.

5. Conclusions and future work

In this paper, we have proposed an approach to the study of network reliability based on combined simulation-interpolation techniques. The basic idea behind this proposal is: if we can obtain the value of the graph's reliability for different edges' operativity probabilities via simulation, then we can interpolate the graph's reliability polynomial.

With these simple concepts in mind, we have initially explored the univariate case, applying the effective interpolation at Chebyshev points. The proposed technique is relatively simple and requires only few lines of code, and it has proven to provide good results, to the extent that there are only few exact solutions that we can compare them with. The method can also be calibrated by intensifying the simulation component, which as we have seen, enables a much higher precision (with the disadvantage of an increased computational cost).

While investigations on how to exactly compute the coefficient of the reliability polynomial in one variable are still ongoing, the bivariate case still remains a nearly unexplored domain. In this paper we have applied our approach to this case too, proposing two methods that combine simulation and interpolation, but where this second component varies: in the first method (BUPI) the interpolation is univariate in each of the two variables independently, whereas the second method (NBPI) explores the Newton bivariate polynomial interpolation. Again, the results are promising and allow us to consider their usefulness in the real world.

A recurrent issue faced during this study was how to test the proposed approach with more realistic graphs such as a given road network. Simulation can also provide a solution to this, although at a high cost: even if the exact reliability polynomial is unknown, could we substitute its value with a simulated reliability for a massive number of simulations? This possibility is intractable with a personal computer, where this additional simulation phase could take multiple days for each value. But we also know that cyberinfrastructures are a common tool for investigators nowadays, and the access to these high-performance computing systems could certainly give an answer to this question.

In times when pandemic has suddenly made modern society stumble and sharpen our defenselessness, the study of networks is even more pressing than it already was when the main focus was the efficiency in different activities from a socioeconomic standpoint (transportation of goods, energy supply, etc.). The consideration of multivariate graphs is not the only open problem whose resolution will bring us closer to real-life scenarios. Other recent developments may consider node failure, ternary networks (where edges could have a third intermediary state between on and off), or diameter-constrained reliability, among other assumptions. The exploration possibilities are boundless, as well as the benefits of investigating them further. And while the exact computation of the reliability polynomial of a graph will be the only the only way to produce perfectly reliable networks, or to allow an optimal degree of control over existing ones, in the meantime, it may also be crucial to explore methods that aim at quasi-optimal solutions. Approximation theory and interpolation techniques are good alternatives, but certainly not the only ones, and other disciplines like metaheuristic optimization would be worth inspecting.

6. Bibliography

- [1] De Boor, C., Ron, A.: *On multivariate polynomial interpolation*. Constructive Approximation volume 6, pages 287–302 (1990)
- [2] Gasca, M., Sauer, T.: *Polynomial interpolation in several variables*. Advances in Computational Mathematics 12(4):377-410 (2012)
- [3] Khorramzadeh, Y.: *Network Reliability: Theory, Estimation, and Applications*. Faculty of the Virginia Polytechnic Institute and State University (2015)
- [4] Moore, E. F., Shannon, C. E.: *Reliable circuits using less reliable relays*. J. Frankl. Inst. 262(3), 191–208 (1956)
- [5] Pérez-Rosés, H.: *Sixty Years of Network Reliability*. Math. Compu. Sci. 12, pages 275–293 (2018)
- [6] Trefethen, L. N.: *Approximation Theory and Approximation Practice*. SIAM (2013)
- [7] Varsamis, D., Karampetakis, N., Mastorocostas, P.: *An Optimal Bivariate Polynomial Interpolation Basis for the Application of the Evaluation-Interpolation Technique*. Appl. Math. Inf. Sci. 8, No. 1, 117-125 (2014)
- [8] Varsamis, D., Karampetakis, N.: *On the Newton bivariate polynomial interpolation with applications*. Multidimensional Systems and Signal Processing 25(1):179-209 (2014)

7. Annexes

Code for section 3.4:

```

%* This is the assessment of the univariate method for the graph K_4-{e}.
%*
%* For different values of M, the obtained approximated reliability
%* is compared against the exact value of the graph's reliability.
%*
%* The assessment will be based on the relative difference
%* and the computation time of the simulation-interpolation step.

%* Presentation of results
fprintf('number of simulations (log_10) \n');
fprintf('elapsed time after simulations-interpolation, in seconds \n');
fprintf('Relative difference for a random value of p \n\n');

%* Graph's static values
n = 4; m = 5;

%* Interpolation point (Chebyshev)
p_0 = 0.5;

%* Adjacency matrix
AdjMat = zeros(n,n); AdjMat(1,2)= p_0; AdjMat(1,3)= p_0; AdjMat(2,3)= p_0; AdjMat(2,4)= p_0; AdjMat(3,4)= p_0;

%* Simulation-interpolation method
for expnnt=1:6
    exptime = 0;
    RD = 0;
    M = 10^expnnt;
    numtests=10;
    for a=1:numtests

        Rel = 0;
        tic;
        parfor i=1:M

            A = AdjMat;
            for j=1:n
                for k=j+1:n
                    if rand < A(j,k)
                        A(j,k) = 1;
                    else
                        A(j,k) = 0;
                    end
                    A(k,j) = A(j,k);
                end
            end

            G = graph(A);
            adj = dfsearch(G,1);
            if size(adj,1) < n
                else
                    Rel = Rel + M^(-1);
                end
            end

            alpha = [p_0^n p_0^m; 1 1]\[Rel-8*p_0^3; 1-8];
            exptime = exptime + toc;
            p_r = rand;
            realRel = 4*p_r^5 - 11*p_r^4 + 8*p_r^3;
            apprRel = alpha(2)*p_r^5 + alpha(1)*p_r^4 + 8*p_r^3;
            if apprRel < 0
                apprRel = 0;
            elseif apprRel > 1
                apprRel = 1;
            end
            if realRel == 0
                fprintf('GHOST');
            else
                RD = RD + abs(realRel-apprRel)/realRel;
            end
        end
    end

    exptime = exptime/numtests; RD = RD/numtests;
    fprintf('%d \n',expnnt); fprintf('%1.4e \n',exptime); fprintf('%1.4e \n',RD); fprintf('\n');
end

```

Code for section 4.4:

```

%* This is the assessment of the bivariate methods for the graph K_4-{e}.
%*
%* For each method, and for different values of M, the obtained approximated
%* reliability is compared against the exact value of the graph's reliability.
%*
%* The assessment will be based on the relative percentage differences
%* and the computation time of the entire method.

```

```

%* Presentation of results
fprintf('number of simulations (log10) \n');
fprintf('elapsed time after simulations, in seconds \n');
fprintf('Relative difference for a random value of p \n\n');

%* Graph's static values
n = 4; mp = 3; mq = 2; m = mp + mq; k1 = max([0 n-1-mq]); k2 = max([0 n-1-mp]); N_L = 8;

%* Interpolation points (Chebyshev)
p = zeros(1,mp-k1+1); q = zeros(1,mq-k2+1);
for i=0:(mp-k1)
    p(mp-k1-i+1) = 0.5*cos(i*pi/(mp-k1+1))+0.5;
end
for j=0:(mq-k2)
    q(mq-k2-j+1) = 0.5*cos(j*pi/(mq-k2+1))+0.5;
end

%* Pre-Adjacency matrices
PreAp = zeros(n,n); PreAq = zeros(n,n); PreAp(1,2)=1; PreAq(1,3)=1; PreAq(2,3)=1; PreAp(2,4)=1; PreAp(3,4)=1;

%* Tests and results for both methods
numtests = 10;
exptime1 = zeros(1,6); exptime2 = zeros(1,6); RD1 = zeros(1,6); RD2 = zeros(1,6);

for a=1:numtests
    p_r = rand; q_r = rand;
    realRel = 4*p_r^3*q_r^2-5*p_r^3*q_r+p_r^3-6*p_r^2*q_r^2+5*p_r^2*q_r+2*p_r*q_r^2;
    fprintf('Test %d of %d: Reference random value done.\n',a,numtests);
    for expnnt=1:6
        %* Reliability assessment at the interpolation points
        %* through SIMULATION:
        simRel = zeros(mp-k1+1,mq-k2+1);
        M = 10^expnnt;
        tic;
        for i=1:(mp-k1+1)
            for j=1:(mq-k2+1)
                AdjMat=(PreAp*p(i))+(PreAq*q(j));
                rrr = 0;
                parfor iter=1:M
                    A = AdjMat;
                    for iT=1:n
                        for jIt=iT+1:n
                            if rand < A(iT,jIt)
                                A(iT,jIt) = 1;
                            else
                                A(iT,jIt) = 0;
                            end
                            A(jIt,iT) = A(iT,jIt);
                        end
                    end
                    G = graph(A);
                    adj = dfsearch(G,1);
                    if size(adj,1) < n
                        rrr = rrr + M^(-1);
                    end
                end
                simRel(i,j) = simRel(i,j) + rrr;
            end
        end
        time = toc;
        exptime1(expnnt) = exptime1(expnnt) + time*(mp-k1+1+mq-k2)/((mp-k1+1)*(mq-k2+1));
        exptime2(expnnt) = exptime2(expnnt) + time;
        fprintf('Test %d of %d: Simulation done.\n',a,numtests);

        %* BDIMENSIONAL UNIVARIATE POLYNOMIAL INTERPOLATION:

        %* Allocating space for the system Ax=B
        %* and building the first and last rows as per equations (4.8), (4.9)
        tic;
        B=zeros(1+mp-k1+mq-k2+3,1); B(1)=N_L; B(1+mp-k1+mq-k2+3)=1;
        A=zeros(1+mp-k1+mq-k2+3,(mp-k1+1)*(mq-k2+1));
        stat1=round((mp-k1+1)/2); statj=round((mq-k2+1)/2);
        statp=p(stat1); statq=q(statj);
        iA=1;
        for i=k1:mp
            for j=k2:mq
                if i+j==n-1
                    A(1,iA)=1;
                end
                iA=iA+1;
            end
        end
        A(1+mp-k1+mq-k2+3,:) = ones(1,(mp-k1+1)*(mq-k2+1));

        %* Univariate interpolation with fixed q
        f=[];
        for i=1:(mp-k1+1)
            f=[f simRel(i,statj)];
        end
        f=f./(p.^k1);
        coeffsA = flip(polyfit(p,f,mp-k1));
        for iter=1:mp-k1+1
            B(1+iter)=coeffsA(iter);
            iA=mp*(iter-1)+1;
            i=k1+iter-1;
            for j=k2:mq
                if i+j>=n-1
                    A(1+iter,iA)=statq^j;
                end
            end
        end
    end
end

```

```

        iA=iA+1;
    end
end

%* Univariate interpolation with fixed p
f=[];
for j=1:(mq-k2+1)
    f=[f simRel(stati,j)];
end
f=f./(q.^k2);
coeffsB = flip(polyfit(q,f,mq-k2));
for iter=1:mq-k2+1
    B(1+(mp-k1+1)+iter)=coeffsB(iter);
    iA=iter;
    j=k2+iter-1;
    for i=k1:mp
        if i+j>n-1
            A(1+(mp-k1+1)+iter,iA)=statp^i;
        end
        iA=iA+mp;
    end
end

x=A\B;
apprRel = 0; e1=k1;
for i=1:(mp-k1+1)
    e2=k2;
    for j=1:(mq-k2+1)
        apprRel = apprRel + x((i-1)*mp+j)*p_r^e1*q_r^e2;
        e2 = e2+1;
    end
    e1 = e1+1;
end
exptime1(expnnt) = exptime1(expnnt) + toc;
if apprRel < 0
    apprRel = 0;
elseif apprRel > 1
    apprRel = 1;
end
if realRel == 0
    fprintf('GHOST');
else
    RD1(expnnt) = RD1(expnnt) + abs(realRel-apprRel)/realRel;
end

%* NEWTON BIVARIATE POLYNOMIAL INTERPOLATION:

%* Sequence of matrices P, starting with P(0) where the minimum exponent
%* factors have been removed
tic;
P = zeros(mp-k1+1,mq-k2+1,mp-k1+1);
for i=1:(mp-k1+1)
    for j=1:(mq-k2+1)
        P(i,j,1)=(simRel(i,j)/(p(i)^k1))/(q(j)^k2);
    end
end
for k = 2:mp-k1+1
    for i = 1:mp-k1+1
        for j = 1:mq-k2+1
            if and(i<k , j<k)
                P(i,j,k) = P(i,j,k-1);
            elseif and(i>=k , j>=k)
                P(i,j,k) = ...
                (P(i,j,k-1)+P(i-1,j-1,k-1)-P(i-1,j,k-1)-P(i,j-1,k-1)) / ...
                ((p(i)-p(i-k+1))*(q(j)-q(j-k+1)));
            elseif and(i<k , j>=k)
                P(i,j,k) = (P(i,j,k-1)-P(i,j-1,k-1))/(q(j)-q(j-k+1));
            elseif and(i>=k , j<k)
                P(i,j,k) = (P(i,j,k-1)-P(i-1,j,k-1))/(p(i)-p(i-k+1));
            end
        end
    end
end

x = sym('x'); y = sym('y');
X = sym('X',[1 mp-k1+1]); Y = sym('Y',[mq-k2+1 1]);
X(1) = 1; Y(1) = 1;
for i = 2:mp-k1+1
    X(i) = expand(X(i-1)*(x-p(i-1)));
end
for j = 2:mq-k2+1
    Y(j) = expand(Y(j-1)*(y-q(j-1)));
end
Rel = (X*P(:, :,mp-k1+1))*Y;
[coefficients, symTerms] = coeffs(Rel, [y x], 'All');

apprRel = 0;
for i=1:(mp-k1+1)
    for j=1:(mq-k2+1)
        cc = eval(coefficients(mq-k2+2-j,mp-k1+2-i));
        apprRel = apprRel + cc*p_r^(i-1+k1)*q_r^(j-1+k2);
    end
end
exptime2(expnnt) = exptime2(expnnt) + toc;
if apprRel < 0
    apprRel = 0;
elseif apprRel > 1
    apprRel = 1;
end
end

```



```

        if realRel == 0
            fprintf('GHOST');
        else
            RD2(expnnt) = RD2(expnnt) + abs(realRel-apprRel)/realRel;
        end

        fprintf('Test %d of %d: finished testing 10^%d.\n\n',a,numtests,expnnt);
    end
end

exptime1 = exptime1/numtests;
exptime2 = exptime2/numtests;
RD1 = RD1/numtests;
RD2 = RD2/numtests;

fprintf('Results BUPI: exponent, time, RD:\n');
for expnnt=1:6
    fprintf('%d \n',expnnt); fprintf('%1.4e \n',exptime1(expnnt)); fprintf('%1.4e \n',RD1(expnnt)); fprintf('\n');
end
fprintf('\nResults NBPI: exponent, time, RD:\n');
for expnnt=1:6
    fprintf('%d \n',expnnt); fprintf('%1.4e \n',exptime2(expnnt)); fprintf('%1.4e \n',RD2(expnnt)); fprintf('\n');
end

```