RT&A, No 4 (89)
Volume 20, December 2025

AN EFFICIENT SINGLE VARIABLE INVERSION ALGORITHM FOR DERIVING RELIABILITY BOUNDS IN STORAGE SYSTEMS

Emad K. Mutar*[1], Zahir A. H. Hassan[2]

•

[1] Department of Mathematics, College of Edu. for Pure Sciences, University of Babylon
Babylon, 51005, Iraq
edu922.emad.kareem@student.uobabylon.edu.iq

[2] Department of Mathematics, College of Edu. for Pure Sciences, University of Babylon
Babylon, 51005, Iraq
zahir.a.h.hassan@uobabylon.edu.iq

Abstract

Ensuring reliability in data survivability schemes for storage systems is challenging due to limited resources. A key part of reliability analysis is selecting a suitable mathematical representation, such as the structure-function. The structure-function maps the overall system state based on the states of its components and can effectively describe systems of various complexities. Many techniques and algorithms depend on identifying all minimal path sets (MPSs) and minimal cut sets (MCSs) to construct the structure and bounds in two-terminal systems. The upper and lower bounds are essential for demonstrating critical states in the system and for the reliability analysis of storage systems. This paper presents three mathematical algorithms for analyzing storage system reliability. The first two algorithms identify MPSs and MCSs using the connection and adjacency matrices of the system's graph. The third algorithm determines critical states of the reliability function by using the single-variable inversion (SVI) method for the MPSs and MCSs, helping to establish the system's reliability bounds. The analysis focuses on finding the greatest lower bound (GLB) and the least upper bound (LUB) for reliability. Storage systems are used to demonstrate the computation of MPSs and MCSs and reliability bounds. These algorithms effectively identify the MPSs and MCSs, calculate their bounds, and visually and numerically represent the results.

Keywords: Minimal paths, minimal cuts, system reliability, sum-of-disjoint products technique, reliability bounds, storage system.

I. Introduction

The structure function is a valuable tool that defines the relationship between a system's performance and the condition of its components. Modeling a repairable system with multiple components in a stationary state provides an important understanding of system reliability. Notably, the structure function can be interpreted as a Boolean function, which has the advantage of offering a time-independent mathematical representation [1-4]. This characteristic allows us to apply established techniques from Boolean algebra, simplifying the evaluation and enhancement of the system's reliability [5-8]. Effective techniques have been developed for reliability analysis using Boolean algebra, which includes defining MPSs and MCSs, calculating frequency characteristics of

DOI: https://doi.org/10.24412/1932-2321-2025-489-318-332

system reliability, and determining importance measures [5, 9]. However, a disadvantage of these methods is that they analyze the system only in a stationary state. Fratta and Montanari [1] introduced the sum-of-disjoint-products (SDP) approach, later refined by Abraham [10]. This technique is commonly used to assess the reliability bounds of two-terminal systems and has been discussed by various researchers [11-16]. Mutar [17] provides a contemporary evaluation of reliability bounds for coherent binary systems, comparing different reliability bounds based on dynamic, mathematical, and efficiency measures. Kumar [18] presents a highly reliable derivative operational matrix method for obtaining approximate solutions to initial and boundary value problems using modified Lucas wavelets. This approach offers greater accuracy and reliability in the results compared to previously available methods. This study leveraging SDP techniques, this approach not only establishes reliable bounds but also incorporates significant advancements through the use of MPSs and MCSs.

The data storage system is a vital part of any information system, as it contains all the essential data needed for its effective operation. Therefore, it is crucial to assess its reliability. Al-Awami and Hassanein [19] delve into the architecture of data survivability systems that leverage decentralized storage within wireless sensor networks. Rusnak et al. [20] introduce a groundbreaking derivative for survival signatures, paving the way for a new method of reliability analysis focused on storage system failures caused by component breakdowns. Additionally, Rusnak et al. [21] present a reliability analysis for data storage that utilizes various types of hard disk drives. The analysis also explores multiple methods of storing data across different disks using a redundant array of independent disks. Mutar [22] addresses the problem of improving the reliability and performance of storage systems by incorporating two types of components with different distributions. Rusnak et al. [23] present a new technique for calculating structural importance measures of storage systems using the survival signature and direct partial logical derivatives.

The objective of this paper is to present three interconnected mathematical algorithms for analyzing the reliability of storage systems. The first and second algorithms aim to identify MPSs and MCSs of the system by utilizing the connection matrix and the adjacency matrix of the system's graph, as well as various algebraic properties. The third algorithm extracts the critical states of the system's reliability function by summing disjoint sets of MPSs and MCSs, which helps determine the system's reliability bounds. The reliability bounds analysis focuses on finding both the GLB and the LUB for system reliability. Furthermore, a reliability bounds calculation for data storage can involve different types of hard disk drives and various techniques for storing data across multiple disks using a redundant array of independent disks, is provided to illustrate the analysis process for the GLB and LUB for system reliability.

The paper is structured as follows: Section II introduces the basic characterizations, formulas, notations, and construction of system reliability. It also presents three interconnected mathematical algorithms for constructing all MPSs and MCSs of the system and using the SVI algorithm for the MPSs and MCSs. Furthermore, mathematical relationships for computing bounds utilizing the disjoint product form are provided. Section III applies the proposed algorithms to calculate all MPSs and MCSs, as well as using the SVI algorithm for the MPSs and MCSs. A simple storage system is used as an example to demonstrate how reliability bounds can be computed, along with numerical comparisons of these bounds. Section IV discusses the application of the proposed algorithms to bridge the design of storage systems. Finally, Section V concludes the paper.

II. Preliminaries

This section introduces the main concept of using a matrix-based approach to generate all MPSs and MCSs of systems. An algorithm is formulated to construct disjoint product terms. Additionally, the system's reliability bounds are determined, focusing on the GLB and LUB for system reliability. The following provides the details.

I. System representation

Consider a system composed of n components that are either functioning in a stable state or being monitored over a specific period. Each component, along with the system operates in one of two essential states: it is either fully functional, represented by "1", or it has failed, represented by "0". A structure function in Eq. (1) defines the vital relationship between the system and the performance of its components.

$$\phi(\boldsymbol{x}) = \phi(x_1, x_2, \dots, x_n): \{0,1\}^n \longrightarrow \{0,1\} \tag{1}$$

In this context, x_i represents the state of component i, where $i = 1,2,\dots,n$. The state vector of all component states is denoted as $\boldsymbol{x} = (x_1, x_2, \dots, x_{\mathrm{n}})$. The reliability of the i-th component, represented in Eq. (2), is defined as the probability that the i-th component is functioning, given by:

$$r_i = \Pr\{x_i = 1\} \tag{2}$$

Moreover, the unreliability of the component, defined as the probability of component failure, is described in Eq. (3) as follows:

$$\overline{r_i} = \Pr\{x_i = 0\} = 1 - r_i \tag{3}$$

The reliability and unreliability of the system, as detailed in Eqs. (4)-(5), are defined using the structure function as follows:

$$R_s = \Pr\{\phi(\boldsymbol{x}) = 1\} \tag{4}$$

$$\overline{R_s} = \Pr\{\phi(\boldsymbol{x}) = 0\} = 1 - R_S \tag{5}$$

There are various algorithms available for calculating system reliability in Eq. (4) and unreliability in Eq. (5). Evaluating the reliability of complex systems requires the use of computationally intensive techniques, such as Path Tracing and Cut methods [5, 9, 24]. MPSs and MCSs are valuable tools in probabilistic reliability analysis. They help identify minimal failure scenarios and are essential for fault tree analysis. Additionally, these techniques enhance the quantification aspect of importance analysis.

Definition 2.1. A path set is a collection of components that must work together to ensure the system operates correctly, especially when the system has experienced a fault. A path set is considered MPSs if no component can be removed without affecting its ability to maintain the system's operational status.

A system is classified as functional when all components within at least one MPSs $(P_1, P_2, \dots, P_{N_P})$ are fully operational, as defined in Definition 2.1. Understanding this concept is essential because it enables us to calculate the system's reliability accurately. The system's reliability can be calculated using the following expression:

$$R_S = \Pr\{P_1 \cup P_2 \cup \dots \cup P_{N_P} \leq \sum_{j=1}^{N_P} \Pr\{P_j\} \tag{6}$$

In this context, N_P represents the total number of MPSs within the system, while P_j (where $j = 1,2,\dots,N_P$) refers to a specific MPS.

Definition 2.2. A cut set is a collection of components whose simultaneous failure leads to a fault in the system, assuming the system was operational beforehand. A cut set is considered MCSs if no component can be removed from it without altering its status as a cut set.

Based on Definition 2.2, the system fails when every component in at least one MCS has failed. Thus, if the system contains N_C of MCSs, the unreliability of the system can be considered as follows:

$$\overline{R_S} = \Pr\{C_1 \cup C_2 \cup \dots \cup C_{N_C} \leq \sum_{j=1}^{N_C} \Pr\{C_j\} \tag{7}$$

where C_j, for $j = 1,2,\dots,N_C$, represents a specific MCS within the system. By utilizing Eqs. (6) and (7), the boundary values for the system's reliability can be effectively estimated as follows:

$$\sum_{j=1}^{N_P} \Pr\{P_j\} \leq R_S \leq 1 - \sum_{j=1}^{N_C} \Pr\{C_j\} \tag{8}$$

When individual system components exhibit high reliability, using the left side of the relation in Eq. (8) is an effective technique for accurately estimating the overall system reliability.

II. Construction of MPSs and MCSs

A system can be represented using the mathematical notation of a graph $G = (V, E)$, where $V = \{1, 2, \dots, m\}$ denotes the set of nodes and $E = \{x_1, x_2, \dots, x_n\}$ denotes the set of edges (or components), which connect the nodes. This graph includes two terminal nodes: a source and a sink. The system graph G supports identifying all MPSs and MCSs between the source and sink nodes.

Definition 2.3. Let G be a graph representing a two-terminal system with vertex set $V = \{1, 2, \dots, m\}$ and edge set $E = \{x_1, x_2, \dots, x_n\}$. The Connection Matrix, denoted as $CM = [e_{ij}]$, is an $m \times m$ matrix with rows and columns indexed by the elements of V. The element e_{ij} is defined as follows:

$$e_{ij} = \begin{cases} 1, & \text{if } i = j, \\ \sum_{x_k \in E} x_k, & \text{if there are edges from node } i \text{ to node } j, \\ 0, & otherwise \end{cases} \tag{9}$$

The element in Eq. (9) can be used to define the CM matrix as follows:

$$CM = \begin{matrix} & \begin{matrix} 1 & 2 & \dots & m \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ m \end{matrix} & \begin{bmatrix} 1 & e_{12} & \cdots & e_{1m} \\ e_{21} & 1 & \cdots & e_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \dots & 1 \end{bmatrix} \end{matrix} \tag{10}$$

The CM matrix in Eq. (10) provides an algebraic representation of a system, assisting in identifying minimal path sets $P_1, P_2, \dots, P_{N_P}$ through algebraic properties. To determine all MPSs of a system based on its graph, the following steps (Algorithm 1) can be utilized:

Step 1: Determine the CM matrix of the system graph's using Definition 2.3.
Step 2: Remove the first column and the last row from the CM matrix to obtain a new matrix (S).

Step 3: Define the system success determinant ($det(S)$), which represents the sum of all MPSs.

$$\begin{aligned}\det(S) &= \sum_{j=1}^{m} (-1)^{i+j} e_{ij} \det(M_{ij}) \\ &= \sum_{j=1}^{m} (-1)^{i+j} e_{ij} \prod_{e_{ij} \in M_{ij}} \det(M_{ij}) \\ &= \sum_{j=1}^{m} (-1)^{i+j} P_j \end{aligned} \tag{11}$$

In this context, M_{ij} in Eq. (11) refers to the minor of the element e_{ij}, which is computed as the determinant of the $(m-1) \times (m-1)$ matrix derived by removing the ith row and the ith column from the CM matrix. On the other hand, the MCSs ($C_1, C_2, \ldots, C_{N_C}$) can be identified using the adjacency matrix (AM).

Definition 2.4: Let G be a graph representing a two-terminal system with vertex set $V = \{1, 2, \ldots, m\}$ and edge set $E = \{x_1, x_2, \ldots, x_n\}$. The adjacency matrix, $AM = [e_{ij}]$, is the m-by-m matrix whose rows and columns are indexed by the elements of V. The element e_{ij} is defined as follows:

$$e_{ij} = \begin{cases} \prod_{x_k \in E} x_k \text{, if there are edges from node } i \text{ to node } j, \\ 0, \quad \quad otherwise \end{cases} \tag{12}$$

The element in Eq. (12) can be utilized to determine the AM matrix as follows:

$$AM = \begin{matrix} \\ 1 \\ 2 \\ \vdots \\ m \end{matrix} \begin{matrix} \begin{matrix} 1 & 2 & \ldots & m \end{matrix} \\ \begin{bmatrix} 0 & e_{12} & \cdots & e_{1m} \\ e_{21} & 0 & \cdots & e_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \ldots & 0 \end{bmatrix} \end{matrix} \tag{13}$$

The AM matrix in Eq. (13) provides an algebraic representation of a system, assisting in identifying MCSs: $C_1, C_2, \ldots, C_{N_C}$ through algebraic properties. To determine all MCSs of a system based on its graph, the following steps (Algorithm 2) can be utilized:

Step 1: Determine the AM matrix of the system graph using Definition 2.4.

Step 2: Multiply all the entries in the first row and multiply all the entries in the last column to obtain two MCSs.

Step 3: Construct a set $\mathcal{S}$ that contains all combinations of columns ranging from order 2 to n-1. This set is crucial for generating every possible MCS, offering valuable alternatives to the MCSs obtained in step 2.

Step 4: Eliminate combinations from set $\mathcal{S}$ that contribute to redundancy or do not qualify as MCSs by adhering to the following rules:

1. If the combination consists only of columns with zero entries in the first row, it should be removed. Keeping these combinations would result in MCSs already included in those generated in step 2.
2. If the combination includes rows with non-zero symbols in the last column, it must also be deleted. These combinations would create MCSs that overlap with the ones generated in step 2.

Step 5: Pick one combination from the set $\mathcal{S}$ identified in Step 4. Then, collect all the link labels from the rows associated with both row 1 and the chosen combination while excluding the columns represented by that combination. This process will allow create a new MCS.

Step 6: Repeat step 5 for all the other combinations in set $\mathcal{S}$.

III. SVI algorithm

The SVI algorithm estimates the probability of multiple events happening at once by identifying the shortest path between them. Boolean algebra is used to determine a system's reliability bounds. Calculating exact reliability in large, complex systems can be difficult due to the numerous calculations required, making approximative techniques necessary. The proposed algorithm (Algorithm 3) for determining system reliability and its bounds is based on equations (7) and (8) as follows:

Step 1: Identify all minimal path sets in the system and arrange them in ascending order, denoted as $P_1, P_2, \dots, P_j, \dots, P_{N_P}$.

Step 2: For the first minimal path set, P_1, establish the initial decision set as $D_{0,1} = P_1$, and calculate its probability, $\lambda_1 = \prod_{x_i \in D_{0,1}} \Pr\{x_i\}$.

Step 3: Set $D_{0,2} = P_2$ and define the complementary set for each minimal path P_j, specifically $P_j - D_{s,i} = \{x\_a \mid x\_a \in P_j \; and \; x_a \notin D_{s,i}\}$, where the indices satisfy $0 \le s < j \le i$ and $a = 1, 2, \dots, n$. Then

- If $P_j - D_{s,i} = \{\emptyset\}$ then $D_{j,i} = \{D_{s,i}\}$ and $\lambda_j = \prod_{x_i \in D_{j,i}} \Pr\{x_i\}$.
- If $x_a \in P_j \; and \; \overline{x_a} \in D_{s,i}$ where $\overline{x_a} = 1 - x_a$ then $P_j - D_{s,i} = \{\emptyset\}$ and $\lambda_j = \prod_{x_i \in D_{s,i}} \Pr\{x_i\}$.
- If $P_j - D_{s,i} = \{x_a\}$ then $D_{j,i} = \{\overline{x_a} D_{s,i}\}$ and $\lambda_j = \prod_{x_i \in D_{j,i}} \Pr\{x_i\}$.
- If $P_j - D_{s,i} = \{x_a, x_b\}$ for $a < b < n$ then $D_{j,r} = \{\overline{x_a} D_{s,i}\}$ and $D_{j,r+1} = \{x_a \overline{x_b} E_{s,i}\}$ then $\lambda_j = \prod_{x_i \in D_{j,r}} \Pr\{x_i\} + \prod_{x_i \in D_{j,r+1}} \Pr\{x_i\}$.

Step 4: Repeat step 3 to form λ_i by comparing $D_{j,i}$ with each P_{j-1}.

Algorithm 3 can form a disjoint sum of the MCSs: $C_1, C_2, \dots, C_{N_C}$ by repeating steps 1 through 5 to achieve β_j for $j = 1, 2, \dots, N_C$.

IV. Reliability bounds of systems

Let λ_j denote the event that all components in the j-th MPS are functioning, meaning that the j-th MPS is operational. The system's operation relies on at least one of the MPSs functioning, which depends on the occurrence of at least one of the events $\lambda_1, \lambda_2, \dots, \lambda_{N_P}$. On the other hand, the event β_j is defined as the failure of all components in the j-th MCSs or the failure of the j-th MCS itself. The failure of the overall system corresponds to the failure of N_C for MCSs, which depends on the occurrence of at least one of the events $\beta_1, \beta_2, \dots, \beta_{N_C}$. The summation of these events represents the system's reliability:

$$R_S = \sum_{j=1}^{N_P} \lambda_j = 1 - \sum_{j=1}^{N_C} \beta_j \tag{14}$$

The SDP technique is a tool for assessing system reliability in Eq. (14), offering an approach to understanding performance. Using MPSs establishes successive lower bounds, while MCSs generate successive upper bounds. These MPSs and MCSs are thoughtfully organized from shortest to longest, capitalizing on a critical principle: when component reliabilities are comparable, shorter MPSs demonstrate a significantly higher probability of success than their longer counterparts. This insight means that the SDP method consistently delivers tighter and more reliable bounds on system performance. The GLB (infimum) of system reliability represents the highest reliability value achievable when at least one set of MPSs fails. In this context, the GLB of system reliability is truly enlightening:

$$R_{GLB} = \sum_{j=1}^{N_P - 1} \lambda_j \tag{15}$$

The LUB (supremum) of system reliability is the smallest value of reliability that corresponds to one MCS. In this case, the LUB of system reliability is:

$$R_{LUB} = 1 - \sum_{j=1}^{N_C-1} \beta_j \tag{16}$$

The techniques for establishing bounds on system reliability are quite effective and direct. Various techniques can generate numerous upper bounds, allowing us to confidently select the LUB (supremum) in Eq. (15) as a solid approximation of the system's reliability. Similarly, all lower bounds are derived, and the GLB (infimum) in Eq. (16) for the system reliability.

III. Simple storage system

I. Model description

Consider a storage system with four components to illustrate how algorithms can be applied to compute reliability bounds. This system consists of two parallel storage units and a communication network, as shown in Fig. 1 [25]. Each unit has two Hard Disk Drives (HDDs) configured in a specific structure known as a Redundant Array of Independent Disks (RAID). The first unit in the upper path contains two HDDs arranged in a RAID-1 configuration, specifically SEAGATE ST6000DX000 and WDC WD60EFRX. In RAID-1, data is written identically to both drives, allowing data to be read from either drive. This arrangement ensures that the system continues to operate successfully as long as at least one drive remains functional. The second unit, located in the lower path, consists of two HDDs configured in a RAID-0 setup, namely SEAGATE ST3000DM001 and HGST HDS5C3030ALA [21, 26]. Unlike RAID-1, RAID-0 provides no data redundancy; instead, the unit's capacity equals the sum of the drives' capacities. Consequently, the entire RAID-0 configuration loses all stored data if one drive fails.

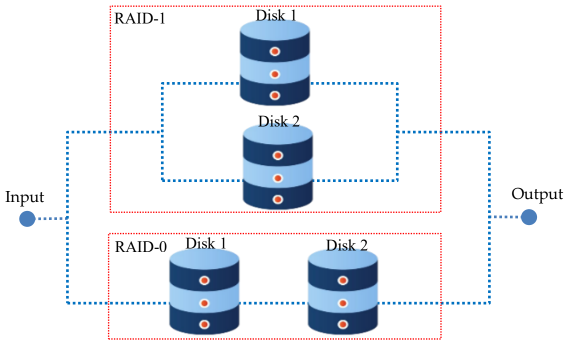

Figure 1. *A schematic diagram of the simple storage system.*

The algorithms outlined in Section II can be utilized to effectively identify the MPSs, minimal cuts, and boundaries of the simple storage system. The simple storage system shown in Fig. 1 is represented by its reliability graph, as depicted in Fig. 2.

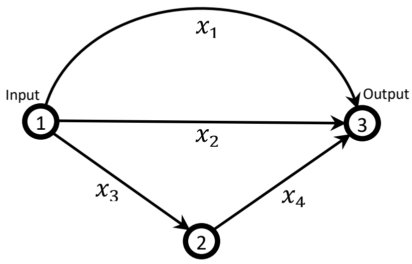

Figure 2. *A graph of the simple storage system in Fig. 1.*

II. The MPSs and MCSs of the storage system.

The MPSs of the storage system depicted in Fig. 2 can be efficiently identified by applying the Algorithm 1 outlined in subsection II. The first step involves constructing the CM matrix as defined in Definition 2.3, specifically tailored for the system shown in Fig. 2, as follows:

$$CM = \begin{matrix} & \scriptstyle 1 & \scriptstyle 2 & \scriptstyle 3 \\ \scriptstyle 1 & 1 & x_3 & x_1 + x_2 \\ \scriptstyle 2 & 0 & 1 & x_4 \\ \scriptstyle 3 & 0 & 0 & 1 \end{matrix} \tag{17}$$

To obtain the S matrix, the first column and last row of the CM matrix in Eq. (17) are removed, and then the determinant of S is calculated as follows:

$$S = \begin{bmatrix} x_3 & x_1 + x_2 \\ 1 & x_4 \end{bmatrix} \tag{18}$$

$$\det(S) = x_1 + x_2 - x_3 x_4 \tag{19}$$

The Eq. (18) and Eq. (19) leads to the identification of three MPSs: $P_1 = \{x_1\}, P_2 = \{x_2\}$ and $P_3 = \{x_3, x_4\}$. Furthermore, the MCSs of the system illustrated in Fig. 2 can be efficiently identified using the Algorithm 2 described in subsection II. The first step is to construct the AM matrix according to Definition 2.4, specifically designed for the system presented in Fig. 2, as explained below:

$$AM = \begin{matrix} & \scriptstyle 1 & \scriptstyle 2 & \scriptstyle 3 \\ \scriptstyle 1 & 0 & x_3 & x_1 x_2 \\ \scriptstyle 2 & 0 & 0 & x_4 \\ \scriptstyle 3 & 0 & 0 & 0 \end{matrix} \tag{20}$$

In the second step of the Algorithm 2 based on Eq. (20), we identify two MCSs: $C_1 = \{x_1 x_2 x_3\}$ and $C_3 = \{x_1 x_2 x_4\}$. The algorithm then stops because there are no further MCSs.

III. The GLB and LUB of the storage system

The reliability function and the GLB of the system depicted in Fig. 2 can be calculated using the MPSs and the third algorithm as follows:

Step 1: Define all MPSs based on order, such that $P_1 = \{x_1\}, P_2 = \{x_2\}$ and $P_3 = \{x_3, x_4\}$.
Step 2: For $P_1 = \{x_1\}$ assume that $D_{0,1} = P_1$

$$\therefore\ \lambda_1 = \prod_{x_i \in D_{0,1}} \Pr\{x_i\} = r_1$$

Step 3: For $P_2 = \{x_2\}$ assume that $D_{0,2} = P_2$

$$\therefore\ P_1 - D_{0,2} = \{x_1\} \Rightarrow \therefore\ D_{1,2} = \left\{\overline{x_1} D_{0,2}\right\} = \{\overline{x_1}, x_2\} \Rightarrow \lambda_2 = \prod_{x_i \in D_{1,2}} \Pr\{x_i\} = \overline{r_1} r_2$$

Step 4: For $P_3 = \{x_3, x_4\}$ assume that $D_{0,3} = P_3$

$$\therefore\ P_1 - D_{0,3} = \{x_1\} \Rightarrow \therefore\ D_{1,3} = \left\{\overline{x_1} D_{0,3}\right\} \Rightarrow D_{1,3} = \{\overline{x_1}, x_3, x_4\}$$

$$\therefore P_2 - D_{1,3} = \{x_2\} \Rightarrow \therefore\ D_{2,3} = \left\{\overline{x_2} D_{1,3}\right\} = \{\overline{x_1}, \overline{x_2}, x_3, x_4\} \Rightarrow \lambda_3 = \prod_{x_i \in D_{2,3}} \Pr\{x_i\} = \overline{r_1}\,\overline{r_2} r_3 r_4$$

Additionally, the reliability function and the LUB of the system illustrated in Fig. 2 can be computed using the MCSs and the third algorithm as follows:

Step 1: Define all MCSs based on order, such that $C_1 = \{\overline{x_1}, \overline{x_2}, \overline{x_3}\}$ and $C_2 = \{\overline{x_1}, \overline{x_2}, \overline{x_4}\}$.
Step 2: For $C_1 = \{\overline{x_1}, \overline{x_2}, \overline{x_3}\}$ assume that $D_{0,1} = C_1$

$$\therefore\ \beta_1 = \prod_{x_i \in D_{0,1}} \Pr\{x_i\} = \overline{r_1}\,\overline{r_2}\,\overline{r_3}$$

Step 3: For $C_2 = \{\overline{x_1}, \overline{x_2}, \overline{x_4}\}$ assume that $D_{0,2} = C_2$

$$\therefore\ C_1 - D_{0,2} = \{\overline{x_3}\} \Longrightarrow \therefore\ D_{1,2} = \{x_3 D_{0,2}\} = \{\overline{x_1}, \overline{x_2}, x_3, \overline{x_4}\} \Longrightarrow \beta_2 = \prod_{x_i \in D_{1,2}} \Pr\{x_i\} = \overline{r_1}\,\overline{r_2}\,r_3\overline{r_4}$$

Based on Eqs. (14)-(16), the GLB, exact value, and LUB for the system reliability in Fig. 2 are provided as follows:

$$R_{GLB} = \lambda_1 + \lambda_2 = r_1 + \overline{r_1} r_2 \tag{21}$$

$$R_S = \lambda_1 + \lambda_2 = 1 - (\beta_1 + \beta_2) = r_1 + \overline{r_1} r_2 + \overline{r_1}\,\overline{r_2} r_3 r_4 \tag{22}$$

$$R_{LUB} = 1 - \beta_1 = 1 - \overline{r_1}\,\overline{r_2}\,\overline{r_3} \tag{23}$$

Assuming that the lifetime distributions of the HDDs follow an exponential distribution with parameter $\lambda = \frac{1}{\text{MTTF}}$, where MTTF represents the mean time to failure. Based on this data, the company estimates individual HDD models' Annualized Failure Rate (AFR) [5, 9, 24, 27]. The AFR demonstrates the probability of a HDD failing during a year of usage, and its connection to MTTF in days is represented by the following equation:

$$MTTF = \frac{-365.25}{Ln(1 - AFR)} \tag{24}$$

The AFRs for the HDDs, evaluated by the company using data from April 2013 to December 2016, are presented in Table 1 [25]. The MTTF values, expressed in days and necessary for calculating the lifetime distributions of the HDDs, were derived from the AFRs by changing the data according to the formula provided in Eq. (24).

Table 1. *The AFR and MTTF values of the HDDs in the storage system shown in Fig. 1.*

Component	Disk name	AFR	MTTF [days]
x_1	SEAGATE ST6000DX000	0.0143	25359
x_2	WDC WD60EFRX	0.0568	62460
x_3	SEAGATE ST3000DM001	0.2672	11750
x_4	HGST HDS5C3030ALA	0.0082	44360

Based on Eqs. (21)-(23) and the MTTFs presented in Table 1, the GLB, exact value, and LUB for the storage system reliability in Fig. 2 are provided as follows:

$$R_{GLB} = -e^{-0.000200t} + e^{-0.000160t} + e^{-0.000039t} \tag{25}$$

$$R_S = e^{-0.001073t} - e^{-0.001034t} - e^{-0.000913t} + e^{-0.000874t} - e^{-0.000200t} + e^{-0.000160t} + e^{-0.000040t} \tag{26}$$

$$R_{LUB} = e^{-0.001051t} - e^{-0.001011t} - e^{-0.000891t} + e^{-0.000851t} - e^{-0.000200t} + e^{-0.000160t} + e^{-0.000039t} \tag{27}$$

The GLB, exact value, and LUB for the reliability of the storage system in Fig. 2 for the period [500, 4000] can be presented in Table 2 and Fig. 3 based on Eqs. (25)-(27).

Table 2. *The R_{GLB}, R_S and R_{LUB} of the storage system in Fig. 1.*

Time [days]	x_1	x_2	x_3	x_4	R_{GLB}	R_S	R_{LUB}
500	0.980476	0.923070	0.653394	0.988792	0.998498	0.999468	0.999479
1000	0.961334	0.852058	0.426924	0.977709	0.994280	0.996667	0.996722
1500	0.942565	0.786509	0.278949	0.966751	0.987738	0.991045	0.991158
2000	0.924162	0.726002	0.182264	0.955915	0.979221	0.982841	0.983008
2500	0.906119	0.670151	0.119090	0.945201	0.969033	0.972519	0.972721
3000	0.888428	0.618596	0.0778128	0.934607	0.957446	0.960541	0.960757
3500	0.871082	0.571007	0.0508424	0.924132	0.944695	0.947294	0.947507
4000	0.854076	0.527079	0.0332201	0.913774	0.930989	0.933084	0.933282

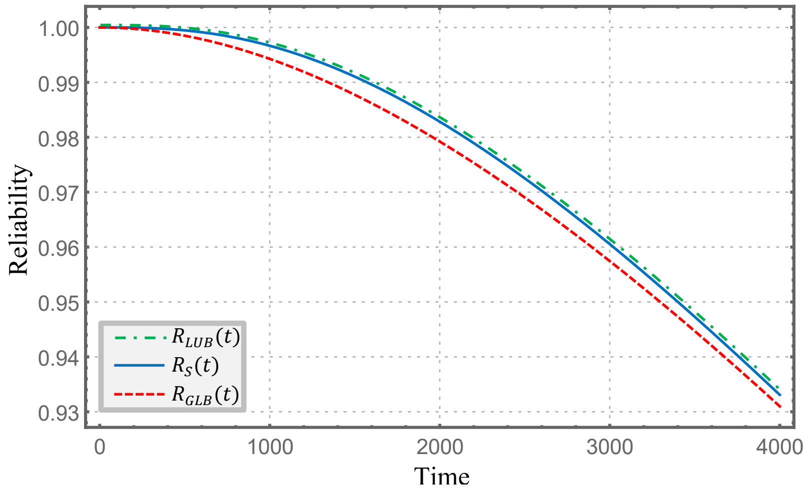

Figure 3. *The GLB, exact value, and LUB for the storage system's reliability for the period [500, 4000].*

Fig. 3 shows the GLB, exact value, and LUB of the storage system. It demonstrates the decline in reliability over time. After 2,500 days, these values fall below 0.969033, 0.972519, and 0.972721.

IV. Bridge topology of storage system

In this section, we demonstrate how to use the proposed algorithms to compute the reliability function, GLB, and LUB for a storage system with a bridge topology consists of five disks, as shown in Fig. 4.

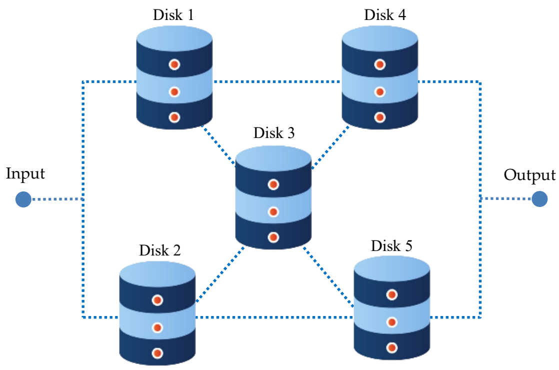

Figure 4. *A schematic diagram of the bridge topology of storage system.*

The storage system illustrated in Fig. 4 can be mathematically represented as a graph $G = (V, E)$, where $V = \{1, 2, 3, 4\}$ and $E = \{x_1, x_2, x_3, x_4, x_5\}$, as shown in Fig. 5. This is a two-terminal, mixed, connected graph that aims to generate all MPSs and MCSs between the source and sink nodes.

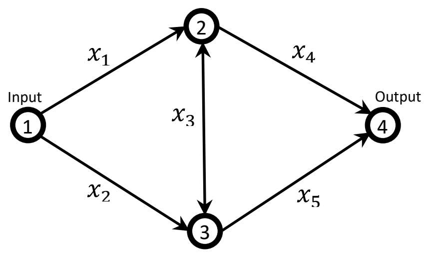

Figure 5. *A graph of the bridge topology of storage system.*

Following Algorithm 1 outlined in subsection 2.2, the MPSs of the system shown in Fig. 5 can be determined using these steps:

Step 1: The CM matrix of the system in Fig. 2 is given in Eq. (28).

$$CM = \begin{matrix} \\ 1 \\ 2 \\ 3 \\ 4 \end{matrix} \begin{matrix} 1 & 2 & 3 & 4 \\ \end{matrix} \begin{bmatrix} 1 & x_1 & x_2 & 0 \\ 0 & 1 & x_3 & x_4 \\ 0 & x_3 & 1 & x_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{28}$$

Step 2: The S matrix of the system in Fig. 2 is given in Eq. (29).

$$S = \begin{bmatrix} x_1 & x_2 & 0 \\ 1 & x_3 & x_4 \\ x_3 & 1 & x_5 \end{bmatrix} \tag{29}$$

Step 3: The determinant of S matrix is given in Eq. (30).

$$\det(S) = x_2x_3x_4 - x_1x_4 + x_1x_3x_5 - x_2x_5 \tag{30}$$

This leads to the identification of four MPSs: $P_1 = \{x_1, x_4\}, P_2 = \{x_2, x_5\}, P_3 = \{x_2, x_3, x_4\}$ and $P_4 = \{x_1, x_3, x_5\}$. Additionally, according to Algorithm 2 detailed in subsection II, the MCSs of the system in Fig. 5 can be identified through the following steps:

Step 1: The AM matrix of the system in Fig. 2 is given in Eq. (31).

$$AM = \begin{matrix} \\ 1 \\ 2 \\ 3 \\ 4 \end{matrix} \begin{matrix} 1 & 2 & 3 & 4 \\ \end{matrix} \begin{bmatrix} 0 & \overline{x_1} & \overline{x_2} & 0 \\ 0 & 0 & \overline{x_3} & \overline{x_4} \\ 0 & \overline{x_3} & 0 & \overline{x_5} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{31}$$

Step 2: The two first MCSs of the system in Fig. 2 is given as follows:
$C_1 = \{\overline{x_1}\ , \overline{x_2}\}$ and $C_2 = \{\overline{x_4}\ , \overline{x_5}\}$

Step 3: The set $\mathcal{S} = \{\{2\}, \{3\}, \{2,3\}\}$ matrix is given as follows:

- For node $\{2\}$, select the first and second rows with remove the second column, then $C_3 = \{\overline{x_2}\ , \overline{x_3}\ , \overline{x_4}\}$.
- For node $\{3\}$, select the first and third rows with remove the third column, then $C_4 = \{\overline{x_1}\ , \overline{x_3}\ , \overline{x_5}\}$.

This leads to the identification of four MCSs: $C_1 = \{\overline{x_1}\ , \overline{x_2}\}, C_2 = \{\overline{x_4}\ , \overline{x_5}\}, C_3 = \{\overline{x_2}\ , \overline{x_3}\ , \overline{x_4}\}$ and $C_4 = \{\overline{x_1}\ , \overline{x_3}\ , \overline{x_5}\}$.

Also, The GLB and exact value for the storage system's reliability depicted in Fig. 5 can be calculated using the MPSs and the third algorithm as follows:

Step 1: Define all MPSs based on order, such that $P_1 = \{x_1, x_4\}, P_2 = \{x_2, x_5\}, P_3 = \{x_2, x_3, x_4\}$ and $P_4 = \{x_1, x_3, x_5\}$.

Step 2: For $P_1 = \{x_1, x_4\}$ assume that $D_{0,1} = P_1$

$$\therefore\ \lambda_1 = \prod_{x_i \in D_{0,1}} \Pr\{x_i\} = r_1 r_4$$

Step 3: For $P_2 = \{x_2, x_5\}$ assume that $D_{0,2} = P_2$

$\therefore\ P_1 - D_{0,2} = \{x_1, x_4\} \Rightarrow \therefore\ D_{1,2} = \{\overline{x_1} D_{0,2}\} = \{\overline{x_1}, x_2, x_5\}$and $D_{2,2} = \{x_1 \overline{x_4} D_{0,2}\} = \{x_1, x_2, \overline{x_4}, x_5\}$

$$\Rightarrow \therefore \lambda_2 = \prod_{x_i \in D_{1,2}} \Pr\{x_i\} + \prod_{x_i \in D_{2,2}} \Pr\{x_i\} = \overline{r_1} r_2 r_5 + r_1 r_2 \overline{r_4} r_5$$

Step 4: For $P_3 = \{x_2, x_3, x_4\}$ assume that $D_{0,3} = P_3$

$\therefore\ P_1 - D_{0,3} = \{x_1\} \Rightarrow \therefore\ D_{1,3} = \{\overline{x_1} D_{0,3}\} = \{\overline{x_1}, x_2, x_3, x_4\}$

$\therefore P_2 - D_{1,3} = \{x_5\} \Rightarrow \therefore D_{2,3} = \{\overline{x_5} D_{1,3}\} = \{\overline{x_1}, x_2, x_3, x_4, \overline{x_5}\} \Rightarrow \therefore \lambda_3 = \prod_{x_i \in D_{2,3}} \Pr\{x_i\} = \overline{r_1} r_2 r_3 r_4 \overline{r_5}$

Step 5: For $P_4 = \{x_1, x_3, x_5\}$ assume that $D_{0,4} = P_4$

$\therefore\ P_1 - D_{0,4} = \{x_4\} \Longrightarrow \therefore\ D_{1,4} = \{\overline{x_4} D_{0,4}\} = \{x_1, x_3, \overline{x_4}, x_5\}$

$\therefore P_2 - D_{1,4} = \{x_2\} \Longrightarrow \therefore\ D_{2,4} = \{\overline{x_2} D_{1,4}\} = \{x_1, \overline{x_2}, x_3, \overline{x_4}, x_5\}$

$\therefore P_3 - D_{2,4} = \{\emptyset\} \Longrightarrow \therefore\ D_{3,4} = D_{2,4} = \{x_1, \overline{x_2}, x_3, \overline{x_4}, x_5\} \Longrightarrow \therefore \lambda_4 = \prod_{x_i \in D_{3,4}} \Pr\{x_i\} = r_1 \overline{r_2} r_3 \overline{r_4} r_5$

Additionally, The LUB and exact value for the storage system's reliability illustrated in Fig. 5 can be computed using the minimal cuts and the third algorithm as follows:

Step 1: Define all MCSs based on order, such that $C_1 = \{\overline{x_1}, \overline{x_2}\}, C_2 = \{\overline{x_4}, \overline{x_5}\}, C_3 = \{\overline{x_2}, \overline{x_3}, \overline{x_4}\}$ and $C_4 = \{\overline{x_1}, \overline{x_3}, \overline{x_5}\}$

Step 2: For $C_1 = \{\overline{x_1}, \overline{x_2}\}$ assume that $D_{0,1} = C_1$

$\therefore\ \beta_1 = \prod_{x_i \in D_{0,1}} \Pr\{x_i\} = \overline{r_1}\, \overline{r_2}$

Step 3: For $C_2 = \{\overline{x_4}, \overline{x_5}\}$ assume that $D_{0,2} = C_2$

$\therefore\ C_1 - D_{0,2} = \{\overline{x_1}, \overline{x_2}\} \Longrightarrow \therefore\ D_{1,2} = \{x_1 D_{0,2}\} = \{x_1, \overline{x_4}, \overline{x_5}\}$ and

$D_{2,2} = \{\overline{x_1} x_2 D_{0,2}\} = \{\overline{x_1}, x_2, \overline{x_4}, \overline{x_5}\} \Longrightarrow \therefore \beta_2 = \prod_{x_i \in D_{1,2}} \Pr\{x_i\} + \prod_{x_i \in D_{2,2}} \Pr\{x_i\} = r_1 \overline{r_4}\, \overline{r_5} + \overline{r_1} r_2 \overline{r_4}\, \overline{r_5}$

Step 4: For $C_3 = \{\overline{x_2}, \overline{x_3}, \overline{x_4}\}$ assume that $D_{0,3} = C_3$

$\therefore C_1 - D_{0,3} = \{\overline{x_1}\} \Longrightarrow \therefore\ D_{1,3} = \{x_1 D_{0,3}\} = \{x_1, \overline{x_2}, \overline{x_3}, \overline{x_4}\}$

$\therefore C_2 - D_{1,3} = \{\overline{x_5}\} \Longrightarrow \therefore\ D_{2,3} = \{x_5 D_{1,3}\} = \{x_1, \overline{x_2}, \overline{x_3}, \overline{x_4}, x_5\}$

$\Longrightarrow \therefore \beta_3 = \prod_{x_i \in D_{2,3}} \Pr\{x_i\} = r_1 \overline{r_2}\, \overline{r_3}\, \overline{r_4} r_5$

Step 5: For $C_4 = \{\overline{x_1}, \overline{x_3}, \overline{x_5}\}$ assume that $D_{0,4} = C_4$

$\therefore\ C_1 - D_{0,4} = \{\overline{x_2}\} \Longrightarrow \therefore\ D_{1,4} = \{x_2 D_{0,4}\} = \{\overline{x_1}, x_2, \overline{x_3}, \overline{x_5}\}$

$\therefore C_2 - D_{1,4} = \{\overline{x_4}\} \Longrightarrow \therefore\ D_{2,4} = \{x_4 D_{1,4}\} = \{\overline{x_1}, x_2, \overline{x_3}, x_4, \overline{x_5}\}$

$\therefore C_3 - D_{2,4} = \{\emptyset\} \Longrightarrow \therefore D_{3,4} = D_{2,4} = \{\overline{x_1}, x_2, \overline{x_3}, x_4, \overline{x_5}\}$

$\Longrightarrow \therefore \beta_4 = \prod_{x_i \in D_{3,4}} \Pr\{x_i\} = \overline{r_1} r_2 \overline{r_3} r_4 \overline{r_5}$

Based on Eqs. (14)-(16), the GLB, exact value, and LUB for the system reliability in Fig. 5 are provided as follows:

$$R_{GLB} = r_1 r_4 + \overline{r_1} r_2 r_5 + r_1 r_2 \overline{r_4} r_5 + \overline{r_1} r_2 r_3 r_4 \overline{r_5} \tag{32}$$

$$R_S = r_1 r_4 + \overline{r_1} r_2 r_5 + r_1 r_2 \overline{r_4} r_5 + \overline{r_1} r_2 r_3 r_4 \overline{r_5} + r_1 \overline{r_2} r_3 \overline{r_4} r_5 \tag{33}$$

$$R_{LUB} = 1 - (\overline{r_1}\, \overline{r_2} + r_1 \overline{r_4}\, \overline{r_5} + \overline{r_1} r_2 \overline{r_4}\, \overline{r_5} + r_1 \overline{r_2}\, \overline{r_3}\, \overline{r_4} r_5) \tag{34}$$

Suppose the components $\{x_1, x_2, x_3, x_4, x_5\}$ of the system in Fig. 5 follow an exponential distribution, with the AFRs of the components being {0.0568, 0.0599, 0.0143, 0.0673, 0.0726}, respectively. Based on Eqs. (32)-(34), the GLB, exact value, and LUB for the reliability of the storage system over the period [500, 4000] can be summarized in Table 3 and illustrated in Fig. 6.

Table 3. *The R_{GLB}, R_S and R_{LUB} of the bridge topology of storage system in Fig. 4.*

Time [days]	x_1	x_2	x_3	x_4	x_5	R_{GLB}	R_S	R_{LUB}
500	0.923070	0.918919	0.980476	0.909032	0.901968	0.978636	0.984657	0.984780
1000	0.852058	0.844412	0.961334	0.826339	0.813546	0.925874	0.943879	0.944623
1500	0.786509	0.775947	0.942565	0.751168	0.733793	0.855015	0.885343	0.887246
2000	0.726002	0.713032	0.924162	0.682836	0.661857	0.775402	0.815820	0.819241
2500	0.670151	0.655219	0.906119	0.620719	0.596974	0.693407	0.740811	0.745887
3000	0.618596	0.602093	0.888428	0.564253	0.538451	0.613216	0.664525	0.671197
3500	0.571007	0.553275	0.871082	0.512924	0.485666	0.537435	0.589997	0.598069
4000	0.527079	0.508415	0.854076	0.466264	0.438055	0.467544	0.519284	0.528477

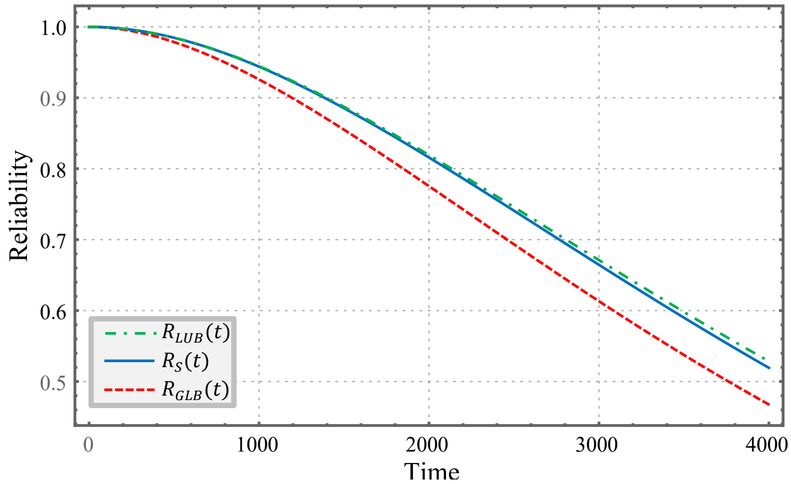

Figure 6. *The GLB, exact value, and LUB for the bridge topology of storage system.*

Fig. 6 illustrates the GLB, exact value, and LUB of the bridge topology of storage system in Fig. 4, highlighting the decline in reliability over time. After 2,500 days, these values drop below 0.693407, 0.740811, and 0.745887. As shown in Table 3, the GLB and LUB obtained using the proposed approach provide a significantly more accurate approximation of the true reliability. To further compare this approach with the Min-Max bounds [28], Edge-packing bounds [29], and Esary-Proschan bounds [30], we refer to Table 4.

Table 4. *Comparison of the GLB and LUB established by the proposed method with other methods.*

Time [days]	R_{GLB}				R_S	R_{LUB}			
	Min-Max [28]	Edge-packing [29]	Esary-Proschan [30]	Propose method		Propose method	Esary-Proschan [30]	Edge-packing [29]	Min-Max [28]
500	0.8391	0.97246	0.984613	0.978636	0.984657	0.984780	0.999084	0.984900	0.991082
1000	0.704088	0.90737	0.943353	0.925874	0.943879	0.944623	0.989827	0.945347	0.96762
1500	0.5908	0.823792	0.883354	0.855015	0.885343	0.887246	0.963792	0.889094	0.933759
2000	0.49574	0.733713	0.81114	0.775402	0.815820	0.819241	0.918574	0.822557	0.892753
2500	0.415975	0.644416	0.732326	0.693407	0.740811	0.745887	0.856855	0.750798	0.84714
3000	0.349045	0.560083	0.65148	0.613216	0.664525	0.671197	0.783751	0.677641	0.798882
3500	0.292883	0.48289	0.572104	0.537435	0.589997	0.598069	0.704758	0.605848	0.74948
4000	0.245758	0.413738	0.496705	0.467544	0.519284	0.528477	0.624637	0.537317	0.70007

For the storage system illustrated in Fig. 5, the proposed approach outperforms the Min-Max bounds and Edge-packing bounds. Its results are close to those of the Esary-Proschan bounds for the GLB. Notably, the proposed approach also surpasses the Min-Max bounds, Edge-packing bounds, and Esary-Proschan bounds for the LUB. Even more impressive is the significant improvement in approximation precision, evidenced by the gap between the exact value for the storage system's reliability and both the GLB and LUB. For instance, at time 2500, the gap between the true reliability and the GLB using the proposed approach is 0.06, while the gap with the LUB is just 0.005. In other words, the proposed approach achieves a more precise approximation of reliability than the methods previously utilized.

V. Conclusion

For large systems, determining exact reliability measures can require a significant amount of time. Often, bounding techniques can provide quicker approximations for a system's reliability or unreliability. This research introduces three mathematical algorithms designed to analyze the reliability of storage systems. The first two algorithms identify all MPSs and MCSs using the connection and adjacency matrices of the system's graph. The third algorithm calculates critical states of the reliability function by summing disjoint sets of MPSs and MCSs, which helps establish the system's reliability bounds. Compared to existing edge-packing methods, the proposed approach offers more precise approximations of network reliability because it identifies and considers a greater number of MPSs. This method demonstrates superior performance relative to the Min-Max bounds and Edge-Packing bounds. Furthermore, its results closely match those of the Esary-Proschan bounds for the GLB. Importantly, the proposed approach also surpasses the Min-Max bounds, Edge-Packing bounds, and Esary-Proschan bounds when evaluating the LUB. These algorithms effectively identify MPSs and MCSs, calculate their bounds, and provide both visual and numerical representations of the results. The computation of MPSs, MCSs, and reliability bounds is demonstrated using storage systems. This research primarily addresses the challenge of expanding networked systems to enhance their reliability. In our future work, we plan to apply the reliability evaluation technique developed in this study to tackle additional issues related to complex networked systems, including vulnerability detection, protection against intentional attacks, and optimization of maintenance strategies. We are also interested in extending the proposed reliability evaluation technique to k-terminal networks, where the MPSs and MCSs are represented as Steiner trees or cuts.

Declarations

Conflict of interest: The authors declare that they have no conflict of interest.
Funding: No funding received for this research.
Consent for publication: This article do not contain studies with human participants or animals by authors.
Data Availability: The no data used to in this article. The authors declare that they have read and approved the final manuscript.

References

[1] Fratta, L. and Montanari, U. (1973). A Boolean algebra method for computing the terminal reliability in a communication network. IEEE Transactions on Circuit Theory, 20(3), 203-211.

[2] Fahmy, M., Ahmed, A., and Khalil, M. (2023). On Reliability: A Mathematical Fault Tree. Reliability: Theory & Applications, 18(1 (72)), 78-85.

[3] Mutar, E. K. and Hassan, Z. A. H. (2025). Survival signature-based structural importance analysis of multi-state system with binary-state components. ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg, 1-18.

[4] Mutar, E. K. and Hassan, Z. A. H. (2025). Joint structure importance of components based on Logic Differential Calculus. Journal of Computational and Applied Mathematics, 117057.

[5] Kuo, W. and Zuo, M. J. (2003). Optimal reliability modeling: principles and applications. John Wiley & Sons.

[6] Hemalatha, G. and Vijayalakshmi, G. (2025). Enhancing sensitivity of MTTF and network reliability stability of k-out-of-n acyclic transmission network. Reliability: Theory & Applications, 20(2), 52-66.

[7] Mutar, E. K. and Hassan, Z. A. H. (2025). Reliability Importance Analysis of a Multi-State System with Binary Components Using Survival Signature. Journal of aerospace technology and management, 17(e1825),

[8] Mutar, E. K. and Hassan, Z. A. H. (2025). Analysis of reliability correlation in the national communication multi-state network of Iraq under time constraints. International Journal of Information Technology,

[9] Aggarwal, K. (2012). Reliability engineering. Springer Science & Business Media.

[10] Abraham, J. A. (1979). An improved algorithm for network reliability. IEEE Transactions on Reliability, 28(1), 58-61.

[11] Ball, M. O. and Provan, J. S. (1988). Disjoint products and efficient computation of reliability. Operations Research, 36(5), 703-715.

[12] Wilson, J. (1990). An improved minimizing algorithm for sum of disjoint products (reliability theory). IEEE Transactions on Reliability, 39(1), 42-45.

[13] Yeh, W.-C. (2007). An improved sum-of-disjoint-products technique for the symbolic network reliability analysis with known minimal paths. Reliability Engineering & System Safety, 92(2), 260-268.

[14] Datta, E. and Goyal, N. K. (2017). Sum of disjoint product approach for reliability evaluation of stochastic flow networks. International Journal of System Assurance Engineering and Management, 8(2), 1734-1749.

[15] Lata, S., Mehfuz, S., Urooj, S., Ali, A., and Nasser, N. (2020). Disjoint Spanning Tree Based Reliability Evaluation of Wireless Sensor Network. Sensors, 20(11), 3071.

[16] Iglesias, R., Pascual-Ortigosa, P., and Sáenz-de-Cabezón, E. (2022). An algebraic version of the sum-of-disjoint-products method for multi-state system reliability analysis. Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation,

[17] Mutar, E. K. (2023). Estimating the reliability of complex systems using various bounds methods. International Journal of System Assurance Engineering and Management, 14(6), 2546-2558.

[18] Kumar, A. (2023). The operational matrix method based on modified lucas wavelets for the approximation solution of initial and boundary value problems. International Journal of Applied and Computational Mathematics, 9(6), 147.

[19] Al-Awami, L. and Hassanein, H. S. (2016). Distributed data storage systems for data survivability in wireless sensor networks using decentralized erasure codes. Computer Networks, 97(113-127.

[20] Rusnak, P., Zaitseva, E., Coolen, F. P., Kvassay, M., and Levashenko, V. (2022). Logic differential calculus for reliability analysis based on survival signature. IEEE Transactions on Dependable and Secure Computing, 20(2), 1529-1540.

[21] Rusnak, P., Sedlacek, P., and Czapp, S. (2023). Reliability Analysis of Data Storage Using Survival Signature and Logic Differential Calculus. In (Ed.), Reliability Engineering and Computational Intelligence for Complex Systems: Design, Analysis and Evaluation (21-35). Springer.

[22] Mutar, E. K. (2024). Reliability Performance Improvement of the Storage System by Using Weibull and Linear-Exponential Distributions. Journal of Advanced Manufacturing Systems, 23(04), 763-784.

[23] Rusnak, P., Zaitseva, E., Levashenko, V., Bolvashenkov, I., and Kammermann, J. (2024). Importance analysis of a system based on survival signature by structural importance measures. Reliability Engineering & System Safety, 243(109814.

[24] Mutar, E. K. (2022). Path Tracing Method to Evaluate the Signature Reliability Function of a Complex System. Journal of aerospace technology and management, 14(

[25] Rusnak, P., Rabcan, J., Kvassay, M., and Levashenko, V. (2019). Time-dependent reliability analysis based on structure function and logic differential calculus. Contemporary Complex Systems and Their Dependability: Proceedings of the Thirteenth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, July 2-6, 2018, Brunów, Poland 13,

[26] Silberschatz, A., Galvin, P. B., and Gagne, G. (2013). Operating system concepts essentials. Wiley Publishing.

[27] Tiwary, A. and Mandloi, R. (2023). An Innovative Approach for Reliability Modeling of HVDC Converter Station. Reliability: Theory & Applications, 18(4 (76)), 1011-1018.

[28] Barlow, R. (1975). Statistical theory of reliability and life testing. Holt,

[29] Colbourn, C. J. (1988). Edge-packings of graphs and network reliability. In (Ed.), Annals of Discrete Mathematics (49-61). Elsevier.

[30] Esary, J. D. and Proschan, F. (1963). Coherent structures of non-identical components. Technometrics, 5(2), 191-209.