Bulletin of Electrical Engineering and Informatics
Vol. 11, No. 3, June 2022, pp. 1568~1576
ISSN: 2302-9285, DOI: 10.11591/eei.v11i3.3509

Evaluating multi-state systems reliability with a new improved method

Yasser Lamalem[1], Soufiane Hamida[2], Yassine Tazouti[1], Oussama El Gannour[2], Khalid Housni[1], Bouchaib Cherradi[2]
[1]Computer research laboratory (L@RI), Faculty of science, University of Ibn Tofail, Kenitra, Morocco
[2]Electrical Engineering and Intelligent Systems (EEIS) Laboratory, ENSET of Mohammedia, Hassan II University of Casablanca, Casablanca, Morocco

Article Info

Article history:

Received Dec 18, 2021
Revised Mar 10, 2022
Accepted May 18, 2022

Keywords:

d-minimal paths
d-MPs
Minimal path
Multi-state system
Optimisation
Reliability

ABSTRACT

The computation of network reliability for a system with many states is an NP-hard issue. Finding all the minimum path vectors (d-MPs) lower boundary points for each level d is one of the few approaches for computing such dependability. This research proposed enhancements to the technique described in Chen's "Searching for d-MPs with rapid enumeration" paper. We propose additional adjustments to the method that creates the flow vector F in this enhancement. This decreases the number of required steps and the temporal complexity of the method. Comparing the newly suggested approach to the old algorithm reveals that the adjustment has increased the enumeration's efficiency and degree of complexity.

Corresponding Author:

Yasser Lamalem
Computer Research Laboratory (L@RI), Faculty of science, University of Ibn Tofail
Avenida de L'Université, Kénitra, Morocco
Email: yasserlamalem@gmail.com

1. INTRODUCTION

The area of machine learning has exceeded the level of performance that can be achieved by humans on a variety of categorization problems with the emergence of deep learning. Concurrently, there is an urgent need to describe and measure the reliability of a model's prediction based on individual samples. This is particularly true when such models are used in the context of safety-critical areas [1]–[3].

The reliability theory aim is to compute the reliability of complex systems from the awareness of the reliabilities of its components. Various methods for computing system reliability have been introduced in the literature. Among these methods we can find cut/path set enumeration [4]–[13], the state enumeration methods [14], [15].

According to the traditional reliability theory, an element can only be in a functioning or failed condition [16]. However, portions of multi-state systems have varying capacity levels under a variety of circumstances. Concerning these systems, we compute the probability that the flow from the origin node towards the sink node is higher than or equal to the specified demand d [17], [18].

The enumeration of all d-minimal pathways (d-MPs) for a certain level d is one of the available approaches [10], [19]–[25] for computing the system dependability of a multi-state system. There are two ways provided in the literature for locating all d-MPs for a certain d. The first one generates d-MPs using direct methods, and the second approach generates d-MPs using pre-enumerated minimal paths (MPs).

Yeh [26] suggested a component-based formulation with three limitations to get d-MP candidates as one of the direct techniques that may generate d-MPs. The second is Ramirez [20], which proposed an additional way for information-sharing mechanisms.

Bai [22] introduced a novel approach that creates the d-MP candidates by merging all the MPs and determining if each d-MP candidate is less than the whole state vector.

In [21], Lin offered a mathematical formulation based on three restrictions for obtaining the candidates for d-MPs from all previously enumerated MPs. Lamalem [25] presented a novel method for listing all d-MPs for all d levels. This method utilizes the network's routes matrix to generate new d-MP candidates from (d-1)-MP d-MP candidates. Chen [27] used a novel approach known as integer-programming [28] to reduce the number of steps required to generate the flow vector F.

Providing new improvements to the Chen [16] method, this study claimed new improvements. These adjustments reduce by a minimum of 60 times the number of steps necessary to get the flow vector F. To demonstrate the effectiveness of the suggested enhancements, we conducted several experiments and compared the results to a well-known method.

The remainder of the paper is structured as follows: The first part provides the method's definitions and underlying assumptions. The stochastic-flow network model is described in section 3. Next, in section 4, we present Chen's algorithm [27] and Lin [21]. Later in section 5, we propose the new improvements to Chen's [27] algorithm. In the end, The efficiency of the new improvements added to the algorithm is compared with Chen [27] and Lin [21].

2. METHODS

2.1. Definitions

For G(A, N, M) is the stochastic-flow network, such that $A=(a_1, ..., a_n)$ defines the group of edges, $N=(n_1, ..., n_m)$ defines the group of nodes, and $M=(M^1, ..., M^n)$ defines the full capacity of each edge a_i.
The path is a group of elements that connect the sources *s* to the sink *t*.
The Minimal Path (MP) is a path that contains no other paths between the source *s* and the terminal *t*.

2.2. Assumptions

The network should specify the fundamental following hypothesis:
- Each state of an arc is statistically independent of another arc.
- The nodes are working perfectly.
- The capacity of every element a_i ranges from $0<1<2...< M^n$.
- The network's flow conforms to the flow-conservation rule.

2.3. Model of stochastic network flow

Let $p_1, p_2, ..., p_z$ represent the collection of MPs from the source *s* to the terminal or sink *t*. The stochastic-flow networking may be represented by two vectors: the flow vector $F=(f_1, f_2, ...f_z)$ and the capacity vector $X=(x_1, x_2, ..., x_n)$, where f_i represents the overall flow of a MP. Moreover, p_i and x_i represents the current capacity of an edge a_i. The vector F is attainable if and only if the following conditions are met:

$$\sum_{j=1}^{z}\left\{f_j \,\middle|\, a_i \in p_j \le M_i,\ i=1, 2, ..., n\right. \tag{1}$$

$$\sum_{j=1}^{z} f_j = d \tag{2}$$

$$f_i \le min\{M^i \,|a_i \in p_j\},\ j=1, 2, ..., m \tag{3}$$

According to constraint (1), the total flow through a_i under F cannot exceed M^i. Constraint (2), states that the entire flow of F equals the supplied demand *d*. The third constraint is that the flow on each MP p_j cannot exceed its maximum capacity. Lemma 1 Any F that fulfills restriction (1) satisfies restriction (3).
Proof 1: If the total of the flows of the several pathways crossing an edge is less than its capacity, then the flow of a specific path is less than the sum of the capacities of the many paths it crosses, indicating the smallest of them.
Let F={F — meets the conditions (1) and (2)}. X = ($x_1, x_2,..., x_n$) is a d-MPs capacity vector for a given d If there exists an F ∈ F such that:

$$x_i = \sum_{j=1}^{z}\{f_j \mid a_i \epsilon p_j\}, i = 1, 2, \dots, n \quad (4)$$

Let $\phi(X)$ represent the flow from node s to sink t, $\phi(X)$ is the capacity or status of the system. Assuming there are L d-MPs, they are denoted by $Z_1, Z_2, Z_3, \dots, Z_L$. Therfore, the possibility that $\phi(X) \geq d$ may be determined as follows:

$$P_r\,(\phi(X)) = P_r\,(\{X \geq Z^1\} \cup \{X \geq Z^2\} \cup \dots . \cup \{X \geq Z^L\}) \quad (5)$$

Ocucurrence $X \geq Z$ denotes $x_i \geq z_i$ for each i, where X and Z are identically sized vectors.

Based on a multistate network with n autonomous nodes. Component i ($1 \leq i \leq n$) has $M_i + 1$ discrete and mutually exclusive states 0, 1,..., M_i. For a specific vector Z whose compenent z_i indicates the position of component i, where:

$$P_r\,(X \geq Z) = \prod_{i=0}^{n} P_r\,(x_i \geq z_i) \quad (6)$$

2.4. Chen and Lin algorithmic improvements

To explain Chen's process, we shall utilize Figure 1 from Chen [27] as a network. For locating all viable solutions to F = $(f_1, f_2, \dots f_z)$, we must apply the constraints (1), (2) and (3).

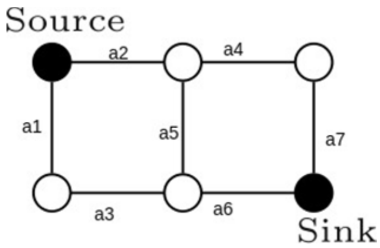

Figure 1. Graph taken from Chen

In Figure 1, there are four minimum paths : $p_1 = (a_1, a_3, a_6)$, $p_2 = (a_2, a_4, a_7)$, $p_3 = (a_2, a_5, a_6)$ and $p_4 = (a_1, a_3, a_5, a_4, a_7)$. They correlate to corresponding F = (f_1, f_2, f_3, f_4), the provided requirement d is 5 and the full capacity for every arc a_i is 5.

From constraint (3), we have :

$$\begin{cases} f_1 \leq 5 \\ f_2 \leq 5 \\ f_3 \leq 5 \\ f_4 \leq 5 \end{cases}$$

From constraint (1), we have :

$$\begin{cases} f_1 + f_4 \leq 5 \\ f_2 + f_3 \leq 5 \\ f_2 + f_4 \leq 5 \\ f_3 + f_4 \leq 5 \\ f_1 + f_3 \leq 5 \end{cases}$$

From constraint (2), we have :

$$f_1 + f_2 + f_2 + f_4 = 5$$

The system of equations to be solved is:

$$\begin{cases} f_1 \le 5, \\ f_2 \le 5, \\ f_3 \le 5, \\ f_4 \le 5, \\ f_1 + f_4 \le 5 \\ f_2 + f_3 \le 5 \\ f_2 + f_4 \le 5 \\ f_3 + f_4 \le 5 \\ f_1 + f_3 \le 5 \\ f_1 + f_2 + f_2 + f_4 = 5. \end{cases}$$

2.4.1. The conventional explicit statement

In [21], Lin's program that utilizes explicit enumeration to explore f_1, f_2, f_3, f_4 is identical to the following code; it employs four loops, and in each loop, the flow f_i begins at 0 and ends at minimum $\{M^i \mid a_i \in p_j\}$, $j = 1, 2, 3, \ldots m$ as a constraint (3)

Lin's Algorithm

```
for (f1=0; f1 ≤ 5; f1++) do
  for (f2=0; f2 ≤ 5; f2++) do
    for (f3=0; f3 ≤ 5; f3++) do
      for (f4=0; f4 ≤ 5; f4++) do
        if f1 + f3 ≤ 5 and f1 + f4 ≤ 5 and f3 + f2 ≤ 5 and f2 + f4 ≤ 5 and f1 + f2 + f3 + f4 = 5
        then
          F = F ∪ {F}
        end if
      end for
    end for
  end for
end for
```

As it can be noticed, Lin's [21] algorithm enters each loop and stops at the last condition to check if it's false or true. If the condition is satisfied, the algorithm adds the flows (f_1, f_2, f_3, f_4) to the set of feasible solutions F, The computation complexity required by running this algorithm using the network of Figure 1 is 6*6*6*6 = 1296 steps (because each flow f_i start from 0 to 5).

2.4.2. Chen's algorithm

Chen's algorithm [27] employs the linear programming approach presented in [28]. The method reorders the constraints in order to decrease the number of required steps. Using the same example from Figure 1, the reorganizations result in:

$$\begin{cases} f_1 \le 5 \\ f_4 \le 5 \\ f_1 + f_4 \le 5 \\ f_3 \le 5, \\ f_1 + f_3 \le 5 \\ f_3 + f_4 \le 5 \\ f_2 \le 5 \\ f_2 + f_3 \le 5 \\ f_2 + f_4 \le 5 \\ f_1 + f_2 + f_2 + f_4 = 5. \end{cases}$$

This strategy of reordering the constraints allows the algorithm to add certain constraints after several iterations, which implies that the algorithm will not enter each iteration as Lin's [21] algorithm does. This strategy used by Chen [28] will decrease the number of steps required to locate all potential answers F.

Chen Algorithm

// $f_1 \le min(a_1, a_3, a_6) = min(5, 5, 5)$, using constraint (1)
for (f_1=0;$f_1 \le 5$;$f_1 ++$) **do**
  for (f_4=0;$f_4 \le 5$;$f_4 ++$) **do**
    if $f_1 + f_4 \le 5$ **then**
      for (f_3=0;$f_3 \le 5$;$f_3 ++$) **do**
        if $f_1 + f_3 \le 5$ and $f_3 + f_4 \le 5$ **then**
          for (f_2=0;$f_2 \le 5$;$f_2 ++$) **do**
            (3)**if** $f_2 + f_3 \le 5$ and $f_2 + f_4 \le 5$ and $f_1 + f_2 + f_3 + f_4 = 5$ **then**
              F =F $\cup$ {F}
            end if
          end for
        end if
      end for
    end if
  end for
end for

Chen's approach [16] requires 546 steps, which is a significant decrease from 1296 to 546 steps by just reordering the requirements. This number can also be decreased by adding new improvements to the algorithm so it will need only 56 steps.

3. PROPOSED IMPROVEMENTS

3.1. Background

Chen's [16] algorithm uses the integer programming problems presented in [28]. Despite that, the algorithm of Chen [16] didn't take advantage of constraint (2), for all the F: f_1 begin from 0 to 5, f_4 begin from 0 to 5, f_3 begin from 0 to 5, and f_2 begin from 0 to 5, which increment the steps needed even with re-ordring the constraints. The purpose of the new enhancements is to appropriately apply constraint (2) to reduce the number of required steps. Constraint (2) states that the entire flow must match the supplied demand d, meaning it cannot exceed d. Consider the following scenario with a demand of 5.

if f_1=3 **then**
if f_4=2 **then**
 $f_1 + f_4 = 5 \le 5$ is true **then**
if f_3=2 **then**
 $f_1 + f_3 = 5 \le 5$ and $f_3 + f_4 = 4 \le 5$ is true **then**
if f2=2 then
 $f_3 + f_2 = 4 \le 5$ and $f_4 + f_2 = 4 \le 5$ are true, but $f_1 + f_2 + f_3 + f_4 = 9 >$ d is false.

By this example, we can deduce that the algorithm traversed all of the loops before reaching the constraint (2). The issue with this approach is that it enters branches that it is not meant to (i.e., we already have the relevant data to decide using constraint (2)). When $f_1 = 3$, the flow f4 should only take values between 0 and 2, not 5, since if $f_4 = 4$, then $f_3 + f_4 = 7$ and 7>d, indicating that the values taken by flow f_4 rely on the value of flow f_3. This example demonstrates that Chen's approach enters branches that cannot provide a solution for which we already have definitive knowledge.

The same applies to f_3, if f_1 takes 3 and f_4 takes 2, f_1 should take only 0 as value, because $f_1 + f_4 = d = 5$. The same thing for f_2. Consequently, the present flow values are dependent on the values of the prior flows. The suggested enhancements determine the current flow maximum capacity that cannot be exceeded by maximizing constraint (2), constraint (3), past flows, and demand d.

Let L_j signify min$\{M^i | a_i \in p_j\}$.

Let V ≡ {F|F are preceding flows}

$$Max\_f_i = min(d - \left(\sum_{j \epsilon V} f_j\right), L_j) \quad (7)$$

Max_f_i is the maximum carrying capacity of the flow f_i .

The final variable after reordering the equations shall only take on a single value that satisfies the requirement (2). We use,

$$\sum_{j=1}^{z} f_j = d \quad (8)$$

Which implies:

$$f_n = d - \sum_{j=1} f_j \tag{9}$$

The last variable after reordering the equations is f_n.

In (7) and (8) will decrease a significant number of steps, and the algorithm will only enter branches that potentially lead to a solution while avoiding time-wasting branches. The constraints derived from (3) will be replaced with constraints derived from (7). Consequently, if these alterations are taken into account, the resultant equation of constraint (2) will be substituted by the result of constraint (3) on the final variable. Consider, for instance, the last loop of algorithm 2:

FOR ($f_2 = 0$; $f_2 \leq 5$; f_2 ++) **do**
IF $f_3 + f_2 \leq 5$ and $f_4 + f_2 \leq 5$ and $f_1 + f_2 + f_3 + f_4 = 5$
F = **F** ∪ {F}
ENDIF
ENDFOR

The following will replace this loop:

$f_2 = 5 - f_1 + f_3 + f_4$ //using (8).
IF $f_2 + f_3 \leq 5$ and $f_2 + f_4 \leq 5$ and $f_2 \leq 5$
F = **F** ∪ {F}
ENDIF

3.2. Illustration

To clarify the new improvements, we took the example used by Chen and Lin. Taking the improvements given previously (section 5), before taking each flow f_i, we compute the $Max\, f_i$ that this flow cannot exceed to avoid many iterations. The new algorithm relating to Figure 1 is given in algorithm 3.

After applying the two proposed in (7) and (8) to the network in Figure 1, when $f_1 = 3$, f_4 will take only the values from 0 to 2 instead of 0 to 5. Another example, when $f_1 = 3$, and $f_4 = 2$, f_3 will take only 0 as value instead the values from 0 to 5. As a result, the number of steps required to locate the flow vector F in this case is 56 instead of 546 of Chen's [16] and 1296 of Lin's [21].

In (8) and (7) reduce a big number of steps compared to algorithm of Chen's [16] because before a flow f_i takes any value, the algorithm makes sure that this value can lead to a feasible solution by using (7). Utilizing these enhancements eliminates needless iterations, hence reducing the number of steps required to construct the flow vector F.

Proposed Algorithm

$Max\_f_1 = min(5 - 0, 5) = 5$
for (f_1=0;$f_1 \leq Max\_f_1$;$f_1 + +$) **do**
  $Max\_f_4 = min(5 - f_1, 5)$
  for (f_4=0;$f_4 \leq Max\_f_4$;$f_4 + +$) **do**
    if $f_1 + f_4 \leq 5$ **then**
      $Max\_f_3 = min(5 - (f_1 + f_4), 5)$
      for (f_3=0;$f_3 \leq Max\_f_3$;$f_3 + +$) **do**
        if $f_1 + f_3 \leq 5$ and $f_3 + f_4 \leq 5$ **then**
          $f_2 = 5 - (f_1 + f_4 + f_3)$
          if $f_2 + f_3 \leq 5$ and $f_2 + f_4 \leq 5$ **then**
            F =**F** ∪ {F}
          end if
        end if
      end for
    end if
  end for
end for

4. RESULTS AND DISCUSSION

To test the efficiency between the new proposed improvements Chen's [16] and Lin [21] algorithms, we used networks below taken from [16]. In Figure 2, we present the five grid networks for all three cases. All algorithms were programmed in C and executed on an HP computer with an Intel Core i7 processor of the second generation and 8 GB of RAM. Table 1 compares the algorithmic stages of Lin's [21], Chen's [16], and the suggested enhancements. Table 2 provides the execution times of the three methods in milliseconds. For all networks, the capabilities of all arcs equal five, and the demand d=5. By adding in (8) and (7), it is evident from Tables 1 and 2 that the new adjustments reduced the number of steps and, therefore, the CPU time required by the algorithm to construct the vector flow F.

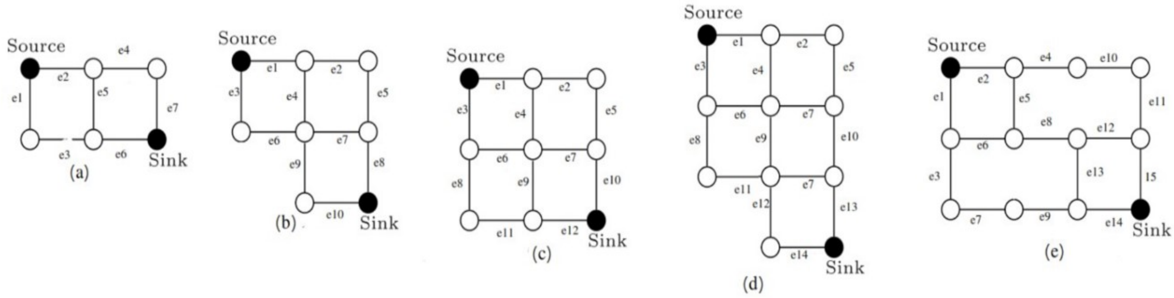

Figure 2. Grid networks example with different nodes

Table 1. Comparison result between the 3 algorithms in terms of steps.

Networks	Algorithms		
	Lin [16]	Chen [11]	Our method
Graph a	1296	546	56
Graph b	279936	9726	462
Graph c	NA	573336	4368
Graph d	NA	121301496	65780
Graph e	NA	693876	11628

Table 2. The execution time of the 3 algorithms in ms.

Networks	Algorithms		
	Lin [16]	Chen [11]	Our method
Graph a	0.051 ms	0.009 ms	0.005 ms
Graph b	1.009 ms	0.11 ms	0.014 ms
Graph c	8821.56 ms	4.355 ms	0.205 ms
Graph d	NA ms	3573.58 ms	8.128 ms
Graph e	NA ms	15.867 ms	ms

5. CONCLUSION

This study provides additional enhancements to Chen's method by taking full use of the constraints that locate the flow vector candidates. As a result, the number of steps required to enumerate the flow vector F is reduced by at least 60 times. Therefore, the amount of time it takes to execute. Lin's and Chen's algorithms were evaluated alongside the newly developed advancements. The findings of the tests indicate that the newly implemented improvements improved the effectiveness of discovering all d-MPs in multi-state networks for a given level d.

REFERENCES

[1] S. Hamida, O. E. Gannour, B. Cherradi, H. Ouajji, and A. Raihani, "Optimization of Machine Learning Algorithms Hyper-Parameters for Improving the Prediction of Patients Infected with COVID-19," in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, Kenitra, Morocco, Dec. 2020, pp. 1–6. doi: 10.1109/ICECOCS50124.2020.9314373.

[2] O. El Gannour *et al.*, "Concatenation of Pre-Trained Convolutional Neural Networks for Enhanced COVID-19 Screening Using Transfer Learning Technique," *Electronics*, vol. 11, no. 1, p. 103, Dec. 2021, doi: 10.3390/electronics11010103.

[3] S. Hamida, O. El Gannour, B. Cherradi, A. Raihani, H. Moujahid, and H. Ouajji, "A Novel COVID-19 Diagnosis Support System Using the Stacking Approach and Transfer Learning Technique on Chest X-Ray Images," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–17, Nov. 2021, doi: 10.1155/2021/9437538.

[4] A. M. Al-Ghanim, "A heuristic technique for generating minimal path and cutsets of a general network," *Computers & Industrial Engineering*, vol. 36, no. 1, Art. no. 1, Jan. 1999, doi: 10.1016/S0360-8352(98)00111-9.

[5] Chin-Chia Jane and Yih-Wenn Laih, "A Practical Algorithm for Computing Multi-State Two-Terminal Reliability," *IEEE Trans. Rel.*, vol. 57, no. 2, pp. 295–302, Jun. 2008, doi: 10.1109/TR.2008.920792.

[6] G. B. Jasmon and O. S. Kai, "A New Technique in Minimal Path and Cutset Evaluation," *IEEE Trans. Rel.*, vol. R-34, no. 2, Art. no. 2, Jun. 1985, doi: 10.1109/TR.1985.5221974.

[7] C. J. Colbourn, "Combinatorial aspects of network reliability," *Ann Oper Res*, vol. 33, no. 1, pp. 1–15, Jan. 1991, doi: 10.1007/BF02061656.

[8] G. Bai, Z. Tian, and M. J. Zuo, "An improved algorithm for finding all minimal paths in a network," *Reliability Engineering & System Safety*, vol. 150, pp. 1–10, Jun. 2016, doi: 10.1016/j.ress.2016.01.011.

[9] K. Housni, "An Efficient Algorithm for Enumerating all Minimal Paths of a Graph," *ijacsa*, vol. 10, no. 1, 2019, doi: 10.14569/IJACSA.2019.0100159.

[10] Y. Lamalem and K. Housni, “New and efficient method to find all minimal paths,” in *2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)*, Marrakech, Morocco, Sep. 2020, pp. 1–4. doi: 10.1109/CommNet49926.2020.9199616.
[11] Y. Lamalem, K. Housni, and S. Mbarki, “New and Fast Algorithm to Minimal Cutsets Enumeration Based on Necessary Minimal Paths,” in *2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT)*, Amman, Oct. 2018, pp. 1–5. doi: 10.1109/ISIICT.2018.8613722.
[12] Y. Lamalem, S. Hamida, K. Housni, N. A. Ali, and B. Cherradi, “A Survey On Recent Algorithms For Multi-state System Exact Reliability Evaluation,” in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Meknes, Morocco, Mar. 2022, pp. 1–6. doi: 10.1109/IRASET52964.2022.9738411.
[13] Y. Lamalem, S. Hamida, K. Housni, A. Ouhmida, and B. Cherradi, “Evaluating Systems Reliability With A New Method Based on Node Cutset,” in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Meknes, Morocco, Mar. 2022, pp. 1–4. doi: 10.1109/IRASET52964.2022.9737956.
[14] A. C. Nelson, J. R. Batts, and R. L. Beadles, “A Computer Program for Approximating System Reliability,” *IEEE Trans. Rel.*, vol. R-19, no. 2, pp. 61–65, May 1970, doi: 10.1109/TR.1970.5216391.
[15] W. Dotson and J. Gobien, “A new analysis technique for probabilistic graphs,” *IEEE Trans. Circuits Syst.*, vol. 26, no. 10, Art. no. 10, Oct. 1979, doi: 10.1109/TCS.1979.1084573.
[16] S.-G. Chen and Y.-K. Lin, “Searching for d -MPs with fast enumeration,” *Journal of Computational Science*, vol. 17, pp. 139–147, Nov. 2016, doi: 10.1016/j.jocs.2016.05.011.
[17] Lamalem, Yasser, “New and Fast Algorithm to Enumerate all Minimal Paths Starting from s and t,” *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 4, Art. no. 4.
[18] X.-Z. Xu, Y.-F. Niu, and C. He, “A Minimal Path-Based Method for Computing Multistate Network Reliability,” *Complexity*, vol. 2020, pp. 1–10, Oct. 2020, doi: 10.1155/2020/8060794.
[19] Y. Lamalem, K. Housni, and S. Mbarki, “Enumeration of the minimal node cutsets based on necessary minimal paths,” *IJ-AI*, vol. 9, no. 2, p. 175, Jun. 2020, doi: 10.11591/ijai.v9.i2.pp175-182.
[20] J. E. Ramirez-Marquez, D. W. Coit, and M. Tortorella, “A generalized multistate-based path vector approach to multistate two-terminal reliability,” *IIE Transactions*, vol. 38, no. 6, pp. 477–488, Jul. 2006, doi: 10.1080/07408170500341270.
[21] Y.-K. Lin, “A simple algorithm for reliability evaluation of a stochastic-flow network with node failure,” *Computers & Operations Research*, vol. 28, no. 13, pp. 1277–1285, Nov. 2001, doi: 10.1016/S0305-0548(00)00039-3.
[22] G. Bai, M. J. Zuo, and Z. Tian, “Search for all d-MPs for all d levels in multistate two-terminal networks,” *Reliability Engineering & System Safety*, vol. 142, pp. 300–309, Oct. 2015, doi: 10.1016/j.ress.2015.04.013.
[23] Y. Wei-Chang, “Fast algorithm for searching d -mps for all possible d.”
[24] Y. Lamalem and K. Housni, “An Improved Algorithm to Search All d-MPs for a Multi-State Systems,” in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, Kenitra, Morocco, Dec. 2020, pp. 1–6. doi: 10.1109/ICECOCS50124.2020.9314374.
[25] Y. Lamalem, K. Housni, and S. Mbarki, “An Efficient Method to Find All d-MPs in Multistate Two-Terminal Networks,” *IEEE Access*, vol. 8, pp. 205618–205624, 2020, doi: 10.1109/ACCESS.2020.3038116.
[26] W.-C. Yeh, “A novel method for the network reliability in terms of capacitated-minimum-paths without knowing minimum-paths in advance,” *Journal of the Operational Research Society*, vol. 56, no. 10, pp. 1235–1240, Oct. 2005, doi: 10.1057/palgrave.jors.2601951.
[27] Y.-K. Lin and S.-G. Chen, “An efficient searching method for minimal path vectors in multi-state networks,” *Ann Oper Res*, Feb. 2019, doi: 10.1007/s10479-019-03158-6.
[28] S. G. Chen, “Efficiency improvement in explicit enumeration for integer programming problems,” in *2013 IEEE International Conference on Industrial Engineering and Engineering Management*, Bangkok, Thailand, Dec. 2013, pp. 98–100. doi: 10.1109/IEEM.2013.6962382.

BIOGRAPHIES OF AUTHORS

Yasser Lamalem was born in Kenitra, Morocco. He received the master’s degree in computer science from Ibn Tofail University, in 2016, where he is currently pursuing the Ph. D. degree in network reliability. His current research interests include multi-state networks reliability and binarystate networks reliability. He can be contacted at email: yasserlamalem@gmail.com.

Soufiane Hamida was born in 1993 at Casablanca-Morocco. He gets master’s degree in educational technology in 2017 from Higher Normal School of Tetouan, Abdelmalek Essaadi University. He is currently an Active Researcher with the Signals, Distributed Systems, and AI Research Laboratory, Hassan II University of Casablanca. He is a Ph. D student since 2017, for the subject of Machine learning methodologies applied for pattern recognition. He Published and was contributed for publication for many research works. His research interests include handwritten recognition characters and words, Parameter’s optimization of machine learning models, Prediction of Patients with Heart Disease, breast cancer, and COVID19 using artificial intelligence techniques. He can be contacted at email: hamida.93s@gmail.com.

Yassine Tazouti is currently associate Professor at School of Management, Computer Science and Telecommunications, Rabat, Morocco. He obtained his PhD in computer science at Ibn Tofail University, Morocco in 2021. He did his First Degree in networks and telecommunications in 2012 at the same institution. He also obtained his master's degree in computer science at Abdelmalek Essaadi university in 2014. His areas of interests are virtual reality, augmented reality, computer vision, virtual reality serious games applications. He can be contacted at email: yassine.tazouti@gmail.com

Oussama El Gannour was born in 1994 at Taza. In 2018, he obtained a Master's Specialized degree in Internet of Things and Mobile Services from the National Higher School of Computer Science and Systems Analysis of Rabat, Mohammed V University. In 2019, he is a PhD student at the Research Laboratory on Electrical Engineering and Intelligent Systems of the Hassan II University of Casablanca. His research interests Contribute to the integration of the Internet of Things and artificial intelligence in the medical field. His research works published include screening Patients with Heart Disease, and COVID19 using AI techniques and IoT. He can be contacted at email: oussama.elgannour@gmail.com.

Khalid Housni received the Master of Advanced Study degree in applied mathematics and computer science, and the Ph.D. degree in computer science from the Ibn Zohr University of Agadir, Morocco, in 2008 and 2012, respectively. He joined the Department of Computer Science, University Ibn Tofail of Kenitra, Morocco, in 2014, where he has been involved in several projects in video analysis and networks reliability. His current research interests include image/video processing, computer vision, machine learning, artificial intelligence, pattern recognition, and networks reliability. He can be contacted at email: housni.khalid@uit.ac.ma

Bouchaib Cherradi was born in 1970 at El Jadida, Morocco. He received the B.S. degree in Electronics in 1990 and the M.S. degree in Applied Electronics in 1994 from the ENSET Institute of Mohammedia, Morocco. He received the DESA diploma in Instrumentation of Measure and Control (IMC) from Chouaib Doukkali University at El Jadida in 2004. He received his Ph.D. in Electronics and Image processing from the Faculty of Science and Technology, Mohammedia. His research interests include applications of Massively Parallel Architectures, Cluster Analysis, Pattern Recognition, Image Processing, Fuzzy Logic systems, Artificial Intelligence, Machine Learning and Deep Learning in Medical and educational data analysis. Dr. Cherradi works actually as an Associate professor in CRMEF-El Jadida. In addition, he is associate researcher member of Signals, Distributed Systems and Artificial Intelligence (SSDIA) Laboratory in ENSET Mohammedia, Hassan II University of casablanca (UH2C), and LaROSERI Laboratory on leave from the Faculty of Science El Jadida (Chouaib Doukali University), Morocco. He is a supervisor of several PhD students. He can be contacted at email: bouchaib.cherradi@gmail.com.