The Relative Isolation Probability of a Vertex in a Multiple-Source Edge-Weighted Graph

Renzo Roel P. Tan*, Kyle Stephen S. See, Jun Kawahara, Kazushi Ikeda, Richard M. de Jesus, Lessandro Estelito O. Garciano, and Agnes D. Garciano

***Abstract*—Various measures that characterize graphs exist in literature. Insights into the properties of a graph as a whole and its components are revealed largely through graph measures, also called graph metrics. In seeking to interpret a consequential edge metric from a vertex-centric perspective, the paper advances an original measure – the relative isolation probability of a vertex. Concisely, the probability of relative isolation pertains to the likelihood of a vertex to be disconnected from all designated source vertices in a graph with probability-weighted edges. A two-step algorithm for efficient calculation is presented and evaluated. Contained within the procedure is a Monte Carlo simulation and the use of a compact data structure called the zero-suppressed binary decision diagram, efficiently constructed through the frontier-based search. The novel measure is then computed for a diverse set of graphs, serving as benchmark for the proposed method. In closing, case studies on real-world networks are performed to ensure the consistency of the experimental with the actual.**

***Index Terms*—frontier-based search, graph measure, Monte Carlo method, probability of relative isolation, zero-suppressed binary decision diagram.**

I. Introduction

THE world operates through networks, thereby establishing the need to study the graphs through which they are represented. On that account, graph measures or graph metrics have long been a domain of interest. Classical metrics such as the connectivity, distance, betweenness, clustering, and reliability polynomial are commonly used to characterize graphs [1]. In addition, there are spectral measures that look into the matrices associated with the graphs [2].

Among the aforementioned are measures that aim to examine whether or not a network would remain to be functioning satisfactorily in the event of damage [3]. In the research, a new graph metric called the probability of relative isolation is introduced. The relative isolation probability reveals how likely it is that a vertex would be disconnected from sources given the failure of some edges. Lifelines such as road networks, water pipelines, power lines, communication systems, *et cetera* and their possible failure during stress are contexts to which the measure is especially applicable. Several advantages to its utility are being able to:

- Know which nodes may require more urgent attention after a destructive episode;
- Have a reasonable basis for the sequence of links for immediate repair;
- Gauge if there are redundancies in the network design and construction;
- Discover prospective node or link groups for further network reinforcement; and
- Estimate possible node locations for the process of adding more network sources.

To compute for the vertex probabilities of relative isolation, a two-step procedure is devised. The first step employs the use of randomness and iteration through conducting a Monte Carlo experiment. The random numbers determine the survival or failure of the edges and consequently, the active components of the graph for each iteration. A check for any source connection is done on the vertices in the second step, having recourse to a compressed data representation known as the zero-suppressed binary decision diagram.

The paper is summarized as follows. A thorough presentation of the relative isolation probability as a novel metric for graphs is contained in the second section. The succeeding two sections review preliminary concepts for efficient implementation, the zero-suppressed binary decision diagram and the frontier-based search. The methodology immediately follows, covering both the algorithm details and the machine specifications. The sixth section discusses the results in two parts – benchmark and real-world figures are supplied. In the seventh section, the practicality of the measure is discussed in greater detail through a case study. The work is concluded in the last section with a recapitulation of the contribution and some recommendations for future research.

Basic notation used throughout are infra. Uniformity with [4] is decided.

V	The set of vertices $\{v_1, v_2, v_3, \ldots, v_{\lvert V\rvert}\}$.
E	The set of edges $\{e_1, e_2, e_3, \ldots, e_{\lvert E\rvert}\}$.

*Author to whom correspondence should be addressed.

Manuscript submitted May 17, 2021; revised November 12, 2021. The study is supported by the Japan Society for the Promotion of Science through the Grants-in-Aid for Scientific Research Program (KAKENHI 18K19821).

R. R. P. Tan is assistant professor at the Mathematical Informatics Laboratory, Division of Information Science, Nara Institute of Science and Technology, Nara Prefecture, Japan and the Department of Mathematics, School of Science and Engineering, Ateneo de Manila University, National Capital Region, Philippines e-mail: rr.tan@is.naist.jp, rrtan@ateneo.edu.

K. S. S. See is engineer at the Augmented Intelligence Pros, Incorporated, National Capital Region, Philippines e-mail: kyle1s@ai-pros.com.

J. Kawahara is associate professor at the Department of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University, Kyoto Prefecture, Japan and the Large Scale Systems Management Laboratory, Division of Information Science, Nara Institute of Science and Technology, Nara Prefecture, Japan e-mail: jkawahara@i.kyoto-u.ac.jp, jkawahara@is.naist.jp.

K. Ikeda is professor at the Mathematical Informatics Laboratory, Division of Information Science, Nara Institute of Science and Technology, Nara Prefecture, Japan e-mail: kazushi@is.naist.jp.

R. M. de Jesus is assistant professor at the Department of Civil Engineering, Gokongwei College of Engineering, De La Salle University, National Capital Region, Philippines e-mail: richard.dejesus@dlsu.edu.ph.

L. E. O. Garciano is professor at the Department of Civil Engineering, Gokongwei College of Engineering, De La Salle University, National Capital Region, Philippines e-mail: lessandro.garciano@dlsu.edu.ph.

A. D. Garciano is associate professor at the Department of Mathematics, School of Science and Engineering, Ateneo de Manila University, National Capital Region, Philippines e-mail: agarciano@ateneo.edu.

G	The graph (V, E) defined by V and E.
S	The set of source vertices $\{s_1, s_2, s_3, \ldots, s_{\lvert S\rvert}\}$.
U	A finite universe $\{x_1, x_2, x_3, \ldots, x_{\lvert U\rvert}\}$.
$\mathcal{F}$	A family of subsets from U.
$\mathcal{D}$	The zero-suppressed binary decision diagram for the family $\mathcal{F}$.
$n(\mathcal{D})$	The number of nodes in $\mathcal{D}$.
$\lvert\mathcal{D}\rvert$	The number of elements represented by $\mathcal{D}$.
$f : A \to B$	A function mapping the set A to the set B.

II. Probability of Relative Isolation

Requisite for the formulation of the metric is the notion of a path in graph theory.

Definition 1 (Path)**.** Given a graph, a path is a sequence $y_1, y_2, \ldots, y_p$ where $y_i = \{v_{i-1}, v_i\}$ is an edge of the graph for $i = 1, 2, \ldots, p$ with vertex $v_i \neq v_j$ of the graph if $i \neq j$. A path from vertex v_0 to vertex v_p is called a v_0-v_p path.

A path is thus a sequence of edges that joins a sequence of vertices which are distinct. The formal definition of the relative isolation probability of a vertex in a given graph is below.

Definition 2 (Probability of Relative Isolation)**.** Consider the graph $G = (V, E)$ and the set of source vertices $S \subset V$. Let the probability of failure of an edge be $\pi(e_i)$ for $i = 1, 2, 3, \ldots, |E|$, with function $\pi : E \to [0, 1]$.

The number of iterations, which is essentially how many times the Monte Carlo experiment is executed, is set to be N. A random number $\rho_k(e_i)$ with function $\rho_k : E \to [0, 1]$ is assigned to each edge e_i on the kth iteration for $k = 1, 2, 3, \ldots, N$. For all $e_i \in E$, an indicator function μ_k is defined to be

$$\mu_k(e_i) = \begin{cases} 1 & \text{if } \rho_k(e_i) > \pi(e_i) \\ 0 & \text{otherwise} \end{cases},$$

assigning either a 1 or a 0 to each edge indicating its survival or failure, respectively, for iteration k.

Subsequently, a subgraph $G_k = (V_k, E_k)$ is produced, with $e_i \in E_k$ if and only if $\mu_k(e_i) = 1$ and $v_j \in V_k$ for $j = 1, 2, 3, \ldots, |V|$ if and only if there exists $e \in E_k$ such that $v_j \in e$ or if $v_j \in S$. More precisely, $E_k = \{e \in E \mid \mu_k(e) = 1\}$ and $V_k = \{v \in V \mid v \in e,\ e \in E_k\} \cup S$ in notation. For all $v_j \in V \setminus S$, a second indicator function is then defined as

$$\lambda_k(v_j) = \begin{cases} 0 & \text{if there exists a path in } G_k \text{ to some } s \in S \\ 1 & \text{otherwise} \end{cases},$$

specifying whether or not each vertex is reachable by a source for iteration k. If $v_j \in S$ then $\lambda_k(v_j) = 0$.

Given the above, the relative isolation probability of a vertex v_j within N instances, denoted by $\Pi_\iota^N(v_j)$, is

$$\Pi_\iota^N(v_j) = \frac{\sum_{l=1}^{N} \lambda_l(v_j)}{N}.$$

III. Zero-Suppressed Binary Decision Diagram

The section shows similar sequencing as [4]. The zero-suppressed binary decision diagram is a graph-based data structure for the efficient storage and handling of families of sets [5]. More formally, consult the definition as follows [6].

Definition 3 (Zero-Suppressed Binary Decision Diagram)**.** Consider a universe U. For $x_k \in U$, $x_i < x_j$ if and only if $i < j$. A zero-suppressed binary decision diagram is a labeled directed acyclic graph satisfying the following properties.

1) There is only one node with indegree 0 called the root.
2) There are only two nodes with outdegree 0 called the 0-terminal and the 1-terminal, denoted by $\bot$ and $\top$, respectively.
3) A nonterminal node has exactly two outgoing arcs labeled by 0 and 1 called the 0-arc and 1-arc, respectively.
4) The destination of the 0-arc and the 1-arc of a nonterminal node is called the 0-child and 1-child, respectively.
5) A nonterminal node is labeled by an element of U.
6) The label of a nonterminal node is strictly smaller than those of its children.

There exists a unique reduced zero-suppressed binary decision diagram with the fewest nodes for a family of concern [6]. Reduction of a diagram in linear time apropos of the number of nodes is seen in [7].

Remark 1. A reduced zero-suppressed binary decision diagram adheres to the two points below.

- There is no node whose 1-child is the 0-terminal.
- There are no distinct nodes that have the same label, 0-child, and 1-child.

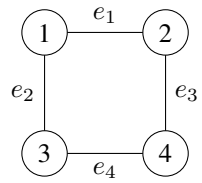
Based on the definition, a family of subsets $\mathcal{F}$ from U may be represented by a single zero-suppressed binary decision diagram $\mathcal{D}$. A path P from the root to the 1-terminal comprising 0-arcs and 1-arcs corresponds to a subset $U' \in \mathcal{F}$ if and only if for all $x \in U'$, there is a node labeled with x whose 1-arc is in P [8]. One proceeds to a theorem that hints at the inherent recursiveness of the diagram [6].

Theorem 1. Let diagram $\mathcal{D}$ correspond to family $\mathcal{F}$. The root e is either a terminal node or a nonterminal node.

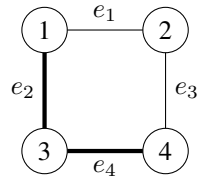
1) If e is the 0-terminal then $\mathcal{F} = \emptyset$, the empty family.
2) If e is the 1-terminal then $\mathcal{F} = \{\emptyset\}$, the family containing only the empty set.
3) If e is nonterminal then it has two children. Let e_0 be the 0-child and e_1 be the 1-child of e. Denote the family with diagram rooted at e_i by $\mathcal{F}_i$. The family $\mathcal{F}$ may then be written as the union $\mathcal{F}_0 \cup \left(\bigcup_{x \in \mathcal{F}_1} x \cup \{e\}\right)$.

Expounding the theorem, the sets in $\mathcal{F}$ that do not contain e are connected to e through its 0-arc. On the other hand, the sets in $\mathcal{F}$ that do contain e are connected to e through its 1-arc. In notation, $\mathcal{F}_0 = \{x \mid x \in \mathcal{F}, e \notin x\}$ and $\mathcal{F}_1 = \{x \setminus \{e\} \mid x \in \mathcal{F}, e \in x\}$.

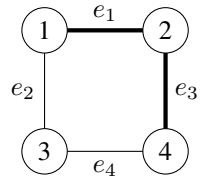
The recursive structure of the zero-suppressed binary decision diagram is revealed. For such a reason, the execution of family operations through the use of the diagram become straightforward [6]. Complexities for several important operations are explicitly stated [9]. Let two diagrams $\mathcal{D}_1$ and $\mathcal{D}_2$ correspond to families $\mathcal{F}_1$ and $\mathcal{F}_2$ with elements from universes U_1 and U_2. Computing for $|\mathcal{D}_1|$ or identically, the number of subsets in $\mathcal{F}_1$, is of time complexity $\mathcal{O}(n(\mathcal{D}_1))$. The enumeration of elements represented by $\mathcal{D}_1$ is of $\mathcal{O}(|\mathcal{D}_1| \cdot |U_1|)$ complexity. Finally, the intersection $\mathcal{F}_1 \cap \mathcal{F}_2$ and the union $\mathcal{F}_1 \cup \mathcal{F}_2$ may each be computed in time complexity $\mathcal{O}(n(\mathcal{D}_1) \cdot n(\mathcal{D}_2))$.



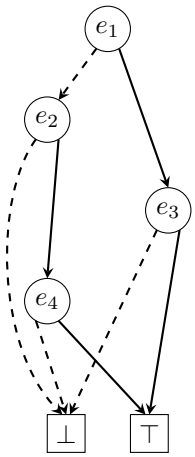
(a) The sample grid.



(b) The first path utilizing edges identified as e_2 and e_4 in the diagram.



(c) The second path utilizing edges identified as e_1 and e_3 in the diagram.



(d) A diagram representing all paths $\{\{e_2, e_4\}, \{e_1, e_3\}\}$ from the first to the fourth vertex in a grid with four vertices.

Fig. 1: Representing paths in a graph using a zero-suppressed binary decision diagram.

For problems under the domain of combinatorics, subset solutions may be represented by a zero-suppressed binary decision diagram [10]. The enumeration of solutions and accordingly, the optimal solution, the number of solutions, the mean and variance of solutions, and other data may be extracted with ease [7]. In a graph-theoretic setting, the edges of a graph may correspond to the items of a universe. Every node in the diagram is labeled with an edge in the graph and its 0-arc and 1-arc indicates the exclusion and inclusion, respectively, of the edge serving as label. As a path from the root node to the 1-terminal of the diagram is a subgraph of the given graph, the entirety of the diagram stands in for a collection of subgraphs. In the case of the paper, the zero-suppressed binary decision diagram is utilized particularly for path enumeration. An illustration explaining how a diagram may represent a set of paths follows.

Consider the square grid of two vertices by two vertices shown in Figure 1a and the possible paths from the first vertex to the fourth vertex shown in Figures 1b and 1c. The zero-suppressed diagram representing all paths is Figure 1d.

All nodes are labeled by a variable number corresponding to an edge and a node identification number for reference in the diagram. A 1-arc is represented by a solid line and a 0-arc is represented by a dashed line. The former and latter correspond to the edge being present or not, respectively. At each level, one proceeds to the 1-child if the edge is included and to the 0-child if the edge is not. Any traversal from the root node to the 1-terminal with symbol $\top$ formed by 1-arcs and 0-arcs, then, is a path in the sample grid. In particular, taking the 0-arc of node e_1, the 1-arc of node e_2, and the 1-arc of node e_4 means taking the path from vertex 1 to vertex 4 through vertex 3; taking the 1-arc of node e_1 and the 1-arc of node e_3 means taking the path from vertex 1 to vertex 4 through vertex 2.

IV. Frontier-Based Search

The frontier-based search is an approach for constructing a zero-suppressed binary decision diagram representing the set of subgraphs satisfying specified constraints [11]. Subgraphs of prescribed types such as paths, matchings, and trees, among others, may be stored in a diagram based on the context of the problem. An outline of the framework for representing a set of paths as laid out in [12] is provided.

The construction of the zero-suppressed binary decision diagram representing the set of all the s-t paths of a given graph is explained below as an example of the frontier-based search. This algorithm is similar to SimPath in [7]. Let $G = (V, E)$ be a weighted undirected graph that is simple and connected. There are to be no multi-edges in G; each element of E is uniquely defined by a 2-subset of V. Let s and t be vertices of V. A subgraph of G is denoted by $G' = (V', E')$, where $E' \subseteq E$ and $V' = \bigcup_{e \in E'} e$. Refining the notation, a union $\bigcup_{i=1}^{j} e_i$ is the set of vertices to which at least one of $e_1, e_2, e_3, \ldots, e_j$ is incident. No vertex of degree 0 may exist in any subgraph. One sets E as the universe for the zero-suppressed binary decision diagram. The elements of E are ordered, with $e_1 < e_2 < e_3 < \cdots < e_{|E|}$.

The diagram is constructed breadth-first, creating and labeling nodes starting from the element in the universe considered to be the smallest. To begin, the root node is labeled with e_1. For $i = 1, 2, 3, \ldots, |E| - 1$, a node labeled e_{i+1} is generated only after nodes labeled e_i are generated. Only the 1-terminal, the 0-terminal, or a node labeled e_{i+1} may serve as destination to both the 0-arc and the 1-arc of any node labeled e_i.

In motivating systematic construction, information on previous edge selection is maintained for vertices which are incident to both a processed edge and an unprocessed edge. Fittingly, the said set of vertices is called the frontier [11]. With $F_0 = F_{|E|} = \emptyset$, the jth frontier is defined in [11] as

$$F_j = \left(\bigcup_{i=1}^{j} e_i \right) \cap \left(\bigcup_{i=j+1}^{|E|} e_i \right)$$

for $j = 1, 2, 3, \ldots, |E| - 1$.

Concurrently, an array $n.\texttt{deg}$ that takes into account subgraph specifications is recorded on each node n. A specified subset of V is mapped to the set of natural numbers by the array. Moreover, to ensure the connectivity of an s-t path, the partition of frontier F_{j-1} is stored in n as $n.\texttt{comp}$. Vertex pairs that belong to the same connected component in G are to be in the same partition in $n.\texttt{comp}$.

Initially, for the root node r, $r.\texttt{deg}[v] = 0$ for all v. For $e = \{u, w\}$, if e is taken then $n.\texttt{deg}[u]$ and $n.\texttt{deg}[w]$ are

incremented by one. For a vertex v, one designates the largest among the indices of incident edges as k. The node v is fixed once e_k has been processed. No further updating is done on $n.\texttt{deg}[v]$ as it is independent of $e_{k+1}, e_{k+2}, e_{k+3}, \ldots, e_{|E|}$. An s-t path is never completed if $n.\texttt{deg}[s]$ or $n.\texttt{deg}[t]$ is not one or $n.\texttt{deg}[v]$ is not zero or two for $v \neq s, t$. In this case, the node becomes $\perp$. If there is no vertex on the frontier that has the same $n.\texttt{comp}$ value as $n.\texttt{comp}[v]$ then the node also becomes $\perp$ because this means that at least two connected components, one including v and another not including v, are generated and are never to be combined.

The node sharing strategy is then hired, merging two nodes n and n' if $n.\texttt{deg}$ is equal to $n'.\texttt{deg}$ and $n.\texttt{comp}$ is equal to $n'.\texttt{comp}$. If a vertex v is in F_{j-1}, $n.\texttt{deg}[v]$ is cached in e_j-labeled node n.

Through node sharing and pruning, the frontier-based search produces a zero-suppressed binary decision diagram. Node generation and information input are simultaneously done towards representing a collection of subgraphs. An unabridged discussion of the technique may be found in [11].

V. Methods

Benchmark configurations from literature and graphs for real networks are each translated into an edge list. Every row is an edge, represented by the two vertices through which it is defined. The source vertices and failure probabilities of the edges are kept in a separate file in order to conveniently create multiple scenarios. Visualization is done in the general-purpose diagramming software yEd[1] with version 3.20.1.

The preliminary processing step is first done through a Monte Carlo simulation, instituting the survival or failure of each edge and generating all G_k for $k = 1, 2, 3, \ldots, N$. Algorithm 1 shows the pseudocode, in which line 11 draws up the random numbers uniformly over the interval $(0, 1)$ for comparison to the relevant probabilities assigned to the edges. The product is a combined input file holding the resulting N instances of the graph.

The computation proper happens in the second step, outlined in Algorithm 2. For each simulation, a zero-suppressed binary decision diagram is constructed for each pair comprising a nonsource vertex v and a source vertex s to determine whether or not there is a path from s to v. The degree constraint dc prescribes v and s as the endpoints of a path, with both vertex degrees being one as designated in Lines 12 and 15). The other vertices are either inner vertices of a path or are not included in the path, with vertex degrees being two or zero as designated in Lines 7, 21, and 26). Using dc, the zero-suppressed binary decision diagram denoted by $paths$ in Line 16 that represents the set of all the paths from s to v is constructed by CONSTRUCTPATHZDD described in the previous section.

Thereafter, a path from s to v used for examining connectivity must only go through available edges. This constraint is represented by the zero-suppressed binary decision diagram of in Line 10. Let E_{avail} be the set of available edges in E. The function CONSTRUCTOMITFAILUREZDD constructs the diagram of representing the set of any subsets of E_{avail}, that is, the power set of E_{avail}. The structure of the zero-suppressed binary decision diagram representing a power set is quite simple; an example is shown in Figure 2. The diagram is constructed outside the for loops of v and s since it is independent from v and s.

The function ZDDINTERSECTION in Line 17 performs the intersection operation on the two zero-suppressed binary decision diagrams, constructing the diagram representing the intersection of the two families of sets represented by the two diagrams [13]. The intersection of $paths$ and of generates the set of paths that use only the available edges as a zero-suppressed binary decision diagram. The existence of a path from the source to the vertex is then inspected by the comparison of zdd and $\perp$ in Line 18. If such a path does not exist then zdd becomes $\perp$, the diagram for the empty set. The check is carried out by the comparison of the two pointers directed to the roots of zdd and $\perp$ in constant time.

[1]Link: https://www.yworks.com/products/yed.

Algorithm 1 MCPrePro

```
1:  G ← readGraph()
2:  S ← readSources()
3:  Assume G = (V, E).
4:
5:  print(|E|, N)
6:  printEOL
7:
8:  for all e ∈ E do
9:      print(e.u, e.v)
10:     for i = 1, ..., N do
11:         if (randUniform(0, 1) ≤ e.p) then
12:             print(1)
13:         else
14:             print(0)
15:         end if
16:     end for
17:     printEOL
18: end for
19:
20: print(|S|)
21: printEOL
22:
23: for all s ∈ S do
24:     print(s)
25: end for
26:
27: printEOL
28: printEOF
```

Note that $e.p$ is the probability that the edge e is alive and $e.u$ and $e.v$ stand for the vertices of the edge e.

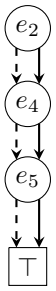


Fig. 2: A zero-suppressed binary decision diagram representing the power set of $\{e_2, e_4, e_5\}$.

Algorithm 2 PIZDD

```
G ← READGRAPH()
S ← READSOURCES()
Assume G = (V, E).

Let isolations and dc be arrays whose indices are vertices.
isolations[v] ← 0 for all v ∈ V
dc[v] ← {0, 2} for all v ∈ V

for sim = 1, ..., N do
    of ← CONSTRUCTOMITFAILUREZDD(sim)
    for v ∈ V − S do
        dc[v] ← {1}
        found ← false
        for s ∈ S do
            dc[s] ← {1}
            paths ← CONSTRUCTPATHZDD(dc)
            zdd ← ZDDINTERSECTION(paths, of)
            if zdd ≠ ⊥ then
                found ← true
            end if
            dc[s] ← {0, 2}
            if found then
                break
            end if
        end for
        dc[v] ← {0, 2}
        if not found then
            isolations[v] ← isolations[v] + 1
        end if
    end for
end for

for v ∈ V do
    PRINT(v, isolations[v]/n)
end for
```

If a valid path from s to v is not found then the algorithm adds 1 to the isolation count of that vertex and moves on to the next vertex; otherwise, it simply moves on to the next vertex. In the end, isolation count of each vertex is divided by the number of simulations to get the probability of relative isolation.

As a comment, keep in mind that source vertices have a relative isolation probability of 0 since there always exists a path to itself. Sources are skipped when running the decision-diagram-based simulation part.

In summary, the algorithm calculates the relative isolation probabilities of the vertices by running a number of randomized simulations, checking if the vertex is disconnected from all sources in each simulation, and getting the percentage of simulations in which the vertex is relatively isolated.

The C++ programming language, coupled with version 9.3.0 of the g++ as compiler, is selected for implementation. Fundamental diagram manipulation is carried out through the TdZdd (https://github.com/kunisura/TdZdd) library documented in [14]. The ZDDLines (https://github.com/renzopereztan/ZDDLines) program serves as foundation for the portion involving the enumeration of paths [15]. For the complete code, refer to the PIZDD (https://github.com/renzopereztan/PIZDD) repository as published online.

Concerning machine specifications, the operating system is the Ubuntu 18.04.4 Long Term Support (Bionic Beaver). The processor is the Intel® Core™ i7-8565U running at 1.80GHz. A memory of 16GB and an NVIDIA® GeForce® MX250 graphics card are also made available for use.

VI. Results

Two segments comprise the results. Outcomes on six sample networks are presented in the first part as benchmark. Graphs with varying structures, seen in Figures 3 to 11, are hand-picked from familiar published work [16], [17]. In the second part, the technique is evaluated on graphs representing real-world lifeline networks. One inspects the Bursa, Hanoi, and Kobe water supply systems [18]–[20].

Sample networks with $6 \leq |V| \leq 40$ and $15 \leq |E| \leq 58$ are first used as setting for the benchmarking experiment. In order, the six are a complete graph, a Petersen graph, two distinctive graphs, a square grid, and a rectangular mesh.

The number of simulations is initialized to 100, 1000, 10000, and 100000. To examine the algorithm correctness, the vertex relative isolation probabilities are computed for edge failure probabilities 0.50, 0.05, and 0.95. The routine is repeated twelve times over per network.

Tables I, II, III, and IV summarize the results. To reiterate, $|E|$ is the number of edges and $|V|$ is the number of vertices. The computation time for each of the two steps is presented both separately and as a sum. For example, $t_{\pi(e),1}$ and $t_{\pi(e),2}$ are costs in seconds for the Monte Carlo instance generator and the decision diagram connectivity test, respectively, and $t_{\pi(e)}$ is the overall time taken. The average time consumed for all set-ups of equal N is $\bar{t}$. In symbols,

$$t_{\pi(e)} = t_{\pi(e),1} + t_{\pi(e),2}$$

and

$$\bar{t} = \frac{t_{\pi_1(e)} + t_{\pi_2(e)} + t_{\pi_3(e)}}{3}.$$

Considering the complexity of an enumerative decision-diagram-based approach, the consumption of time is fair [15], [21]. The total cost ranges from seconds for experiments with 100 iterations to minutes for experiments with 100000 iterations.

The set of probabilities obtained for the graphs when all edges are set to fail half of the time are attached as Appendices A, B, C, D, E, and F. A superscript † indicates a source vertex. Setting the results from the different values for N side by side serve as preview to the convergence of the method.

The procedure is then applied to real-world systems. More specifically, the water distribution networks of three cities – Bursa in Turkey, Hanoi in Vietnam, and Kobe in Japan – are chosen for processing. For the three networks, $12 \leq |V| \leq 32$ and $15 \leq |E| \leq 34$.

The program is run four times for each network, accommodating 100, 1000, 10000, and 100000 instances. The edge probabilities of failure differ per network. Information is sought from references that provide well-justified data. Details on the Bursa city network is found in [22]. Furthermore, a comprehensive analysis of the Hanoi city network may be seen in [19]. Lastly, the Kobe city network is treated in [23].

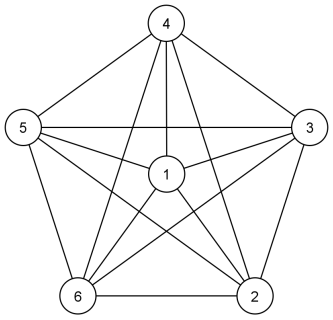


Fig. 3: Network 1.

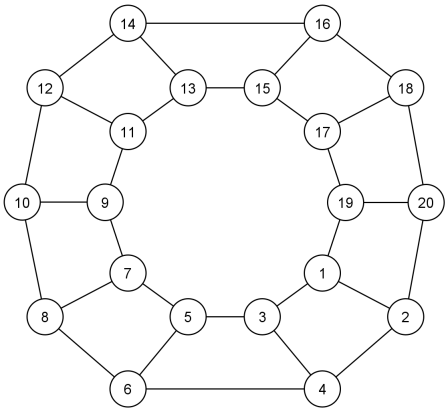


Fig. 4: Network 2.

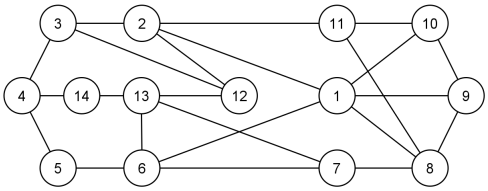


Fig. 5: Network 3.

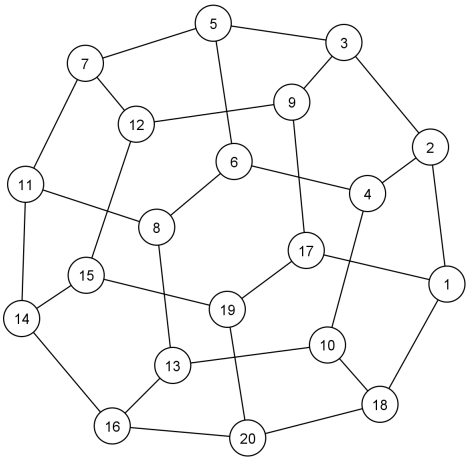


Fig. 6: Network 4.

The results for the lifeline networks are in Tables V, VI, VII, and VIII, following the same format as the previous tables. The vertex probabilities of relative isolation are consistently calculated in reasonable time. An experiment of 100 iterations requires seconds and an experiment of 100000 iterations requires minutes. Likewise, the probabilities are noted as Appendices G, H, and I for reference.

With respect to convergence, it is important to note that the computation hinges on dividing the number of simulations where the vertex is relatively isolated by the total number of simulations. Beyond 100 simulations, therefore, the chances of succeeding simulations making any change larger than a percent is minuscule. Regarding complexity, the following remark is verified.

Remark 2. For a nonsource vertex $v \in V \setminus S$ and source vertex $s \in S$, let $\mathcal{D}_{v,s}^{N}$ be the zero-suppressed binary decision diagram zdd with the most number of nodes considering all N instances. The diagram with the most number of nodes across all constructed diagrams $\mathcal{D}_{v,s}^{N}$ is then denoted by $\mathcal{D}_{\Omega}$. The worst-case complexity of the algorithm is $\mathcal{O}\left(N \cdot |S| \cdot |V \setminus S| \cdot n\left(\mathcal{D}_{\Omega}\right)\right)$.

As an aside, accuracy is further ensured by running the entire pool of experiments multiple times. One remarks that the variability between the initializations is not substantial.

VII. Discussion

A primary application for the probability of relative isolation is under civil engineering. As mentioned in the introduction, the operation of critical lifeline networks is subject to natural hazards. This fact necessitates evaluating the ability of such networks to withstand damage. While there is much evidence to the edge probability of failure being a reliable basis for network resilience metrics, additional details may be obtained from a vertex point of view.

In reality, vertices in a graph representing a lifeline represent points of demand – a household in a water supply system, a corporate office in an electrical grid, a commercial hub in a transport network, and others. By generating the vertex probabilities of relative isolation from the edge probabilities of failure, collective information on edge resilience is translated into vertex resilience. The efficiency and effectiveness of a network in responding to demand under unfavorable circumstances is then better understood.

To exhibit the interpretability of the proposed graph metric, the Kobe water distribution network is investigated as a case study. The representative graph has 20 edges and 15 vertices as configured in Figure 11. Vertices 1 and 6, both marked with an asterisk, serve as sources for the pipeline network. In drawing insight, the given edge probabilities of failure and calculated vertex probabilities of relative isolation are incorporated in a hierarchical layout of the network shown by Figure 12. The upper right quadrant is labeled as Q1 and the lower left quadrant is labeled as Q2 for convenience. For the purpose of visualization, the edges are of different line styles. A solid line means a low failure probability, a dashed line means a moderate failure probability, and a dotted line means a high failure probability. Correspondingly, the vertices are filled with colors from a grayscale scheme. Lighter colors mean lower relative

isolation probabilities; darker colors mean higher relative isolation probabilities.

At a glance, the interpretation of edge failure into vertex relative isolation through simulation proves to be well-grounded. Overall, two contrasting vertex clusters are formed. Found in Q1 is the source vertex 1. As transmission lines boast low failure probabilities, the relative isolation of vertices in the quadrant is not significant. Despite Q2 having a dedicated source – vertex 6, demand points in the quadrant suffer from high relative isolation. The rather high probabilities of edge failure are accurately translated into high probabilities of vertex relative isolation. Observations within the above context follow.

In proactive mitigation, a direct implication of the computed relative isolation probability values is the reinforcement of major resource channels. In Q2, for example, the edge from the vertex 6 to vertex 5 and the edge from vertex 5 to vertex 2, for example, should survive as much as possible. Otherwise, it would look as though Q2 has no source vertex.

Although multiple interquadrant transmission lines exist, this appears to be a futile back-up as all edges share a high likelihood of breakage. In the event of a disaster, it would be unlikely for the source of Q1 to temporarily answer to the demand of Q2. Redundancies in the network, when put in place, should be beneficial.

When a disaster occurs, vertices in Q2 are to be closely monitored. With the vertex probability of relative isolation, one has supplementary information on which points of demand are most vulnerable. In response to the relative isolation of vertices, stand-in sources such as fire trucks and mobile water stations may be deployed.

Observations from the Hanoi water distribution network are similar. Figure 13 is a hierarchical representation of the network, formatted in the same way as the Kobe case study. The right side of the network with respect to the viewer is referred to as S1. The left side of the network with respect to the viewer is referred to as S2. Two water sources are present in S1; however, their service to S2 is prone to disruption. While four links serve as transmission lines from S1 to S2, only one has a low likelihood of failure. Through the use of the probability of relative isolation, redundancy in this network is discovered to be futile. It is clear that vertices in S2 are bound to be isolated from the sources after destructive seismic activity.

VIII. Conclusion

In brief, a novel measure for graphs consisting of edges that have known failure probabilities is put forward. Its definition is accompanied by a demonstration of efficient calculation through a Monte Carlo method integrated with a decision-diagram-based technique. The new probabilistic metric is worked out across selected networks, proving reasonable computational cost. Insights generated by the measure is then discussed in a case study.

For future study, an original consolidated measure for the whole graph based on the vertex probabilities of relative isolation is to be pursued. In regard to probability evaluation, crafting an algorithm that offers less computation time and an implementation on larger networks are suggested.

Appendix A
Probabilities for Network 1

j	$\Pi_\iota^{100}(v_j)$	$\Pi_\iota^{1000}(v_j)$	$\Pi_\iota^{10000}(v_j)$	$\Pi_\iota^{100000}(v_j)$
$1^\dagger$	0.0000	0.0000	0.0000	0.0000
2	0.0600	0.0400	0.0409	0.0430
3	0.0400	0.0430	0.0435	0.0424
4	0.0300	0.0480	0.0446	0.0418
5	0.0500	0.0400	0.0463	0.0415
$6^\dagger$	0.0000	0.0000	0.0000	0.0000

Appendix B
Probabilities for Network 2

j	$\Pi_\iota^{100}(v_j)$	$\Pi_\iota^{1000}(v_j)$	$\Pi_\iota^{10000}(v_j)$	$\Pi_\iota^{100000}(v_j)$
1	0.4500	0.3360	0.3234	0.3294
$2^\dagger$	0.0000	0.0000	0.0000	0.0000
3	0.5300	0.4580	0.4289	0.4365
4	0.3900	0.3720	0.3533	0.3602
5	0.5400	0.5070	0.5087	0.5101
6	0.5200	0.5230	0.5005	0.4944
7	0.5700	0.4910	0.5034	0.4952
8	0.5500	0.5460	0.5164	0.5078
9	0.4000	0.3810	0.3619	0.3572
10	0.4100	0.4680	0.4423	0.4369
$11^\dagger$	0.0000	0.0000	0.0000	0.0000
12	0.3200	0.3330	0.3352	0.3307
13	0.4100	0.3710	0.3590	0.3578
14	0.4600	0.4320	0.4385	0.4366
15	0.5200	0.5220	0.5016	0.4972
16	0.5100	0.4920	0.5151	0.5107
17	0.5100	0.5230	0.5118	0.5125
18	0.4700	0.4950	0.4916	0.4963
19	0.4400	0.4450	0.4391	0.4379
20	0.3300	0.3280	0.3556	0.3551

Appendix C
Probabilities for Network 3

j	$\Pi_\iota^{100}(v_j)$	$\Pi_\iota^{1000}(v_j)$	$\Pi_\iota^{10000}(v_j)$	$\Pi_\iota^{100000}(v_j)$
1	0.1700	0.1520	0.1410	0.1437
2	0.2400	0.3220	0.3113	0.3026
3	0.3600	0.4350	0.4058	0.4007
4	0.2900	0.3490	0.3411	0.3372
$5^\dagger$	0.0000	0.0000	0.0000	0.0000
6	0.2000	0.2140	0.2038	0.2033
7	0.3200	0.3170	0.3067	0.3097
8	0.1500	0.1910	0.1825	0.1880
$9^\dagger$	0.0000	0.0000	0.0000	0.0000
10	0.2300	0.2490	0.2257	0.2311
11	0.3000	0.3180	0.3062	0.3040
12	0.3900	0.4240	0.3964	0.4016
13	0.2800	0.3500	0.3255	0.3252
14	0.4900	0.4950	0.4922	0.4842

APPENDIX D
PROBABILITIES FOR NETWORK 4

j	$\Pi_\iota^{100}(v_j)$	$\Pi_\iota^{1000}(v_j)$	$\Pi_\iota^{10000}(v_j)$	$\Pi_\iota^{100000}(v_j)$
1†	0.0000	0.0000	0.0000	0.0000
2	0.2600	0.3400	0.3321	0.3369
3	0.4500	0.4000	0.4408	0.4421
4	0.4400	0.4440	0.4424	0.4408
5	0.4800	0.4240	0.4407	0.4429
6	0.4900	0.4380	0.4460	0.4423
7	0.3200	0.3490	0.3350	0.3363
8	0.3800	0.3400	0.3455	0.3351
9	0.4500	0.4320	0.4423	0.4409
10	0.4200	0.4250	0.4394	0.4400
11†	0.0000	0.0000	0.0000	0.0000
12	0.4200	0.4390	0.4416	0.4408
13	0.4700	0.4240	0.4493	0.4404
14	0.3200	0.3390	0.3371	0.3344
15	0.4500	0.4280	0.4450	0.4417
16	0.4900	0.4360	0.4473	0.4391
17	0.3800	0.3160	0.3301	0.3367
18	0.2700	0.3240	0.3362	0.3361
19	0.5000	0.4300	0.4494	0.4397
20	0.4300	0.4420	0.4484	0.4387

APPENDIX E
PROBABILITIES FOR NETWORK 5

j	$\Pi_\iota^{100}(v_j)$	$\Pi_\iota^{1000}(v_j)$	$\Pi_\iota^{10000}(v_j)$	$\Pi_\iota^{100000}(v_j)$
1†	0.0000	0.0000	0.0000	0.0000
2	0.3700	0.3370	0.3467	0.3491
3	0.5000	0.4670	0.4983	0.5003
4	0.5700	0.5610	0.5938	0.5925
5	0.7100	0.6560	0.6805	0.6857
6	0.3700	0.3320	0.3422	0.3487
7	0.4400	0.3700	0.3828	0.3892
8	0.4700	0.3990	0.4487	0.4502
9	0.5300	0.4650	0.5078	0.5096
10	0.6000	0.5530	0.5973	0.5917
11	0.5100	0.5050	0.4902	0.5010
12	0.5200	0.4480	0.4468	0.4508
13	0.4800	0.4310	0.4448	0.4445
14	0.5000	0.4170	0.4523	0.4499
15†	0.5100	0.4620	0.5039	0.5000
16	0.6800	0.5910	0.5756	0.5913
17	0.5600	0.5040	0.5005	0.5080
18	0.5000	0.4520	0.4510	0.4499
19	0.3800	0.3590	0.3966	0.3891
20	0.3700	0.3300	0.3556	0.3494
21	0.7900	0.6900	0.6724	0.6854
22	0.6600	0.5910	0.5832	0.5915
23	0.6200	0.4820	0.4992	0.5019
24	0.4000	0.3430	0.3565	0.3489
25	0.0000	0.0000	0.0000	0.0000

APPENDIX F
PROBABILITIES FOR NETWORK 6

j	$\Pi_\iota^{100}(v_j)$	$\Pi_\iota^{1000}(v_j)$	$\Pi_\iota^{10000}(v_j)$	$\Pi_\iota^{100000}(v_j)$
1†	0.0000	0.0000	0.0000	0.0000
2	0.4800	0.4520	0.4328	0.4298
3	0.3800	0.4210	0.4275	0.4303
4	0.5100	0.5430	0.5328	0.5381
5	0.6200	0.6610	0.6453	0.6478
6	0.6400	0.6850	0.6687	0.6747
7	0.7200	0.7840	0.7698	0.7728
8	0.7300	0.7820	0.7790	0.7818
9	0.8000	0.8480	0.8518	0.8522
10	0.7600	0.8550	0.8538	0.8544
11	0.8800	0.9000	0.9020	0.9020
12	0.8500	0.9050	0.9059	0.9030
13	0.9200	0.9250	0.9373	0.9343
14	0.9100	0.9320	0.9359	0.9347
15	0.9400	0.9550	0.9569	0.9546
16	0.9600	0.9500	0.9575	0.9542
17	0.9600	0.9660	0.9667	0.9657
18	0.9600	0.9690	0.9675	0.9663
19	0.9900	0.9760	0.9720	0.9715
20	0.9800	0.9780	0.9725	0.9715
21	0.9900	0.9730	0.9726	0.9718
22	0.9800	0.9820	0.9716	0.9717
23	0.9700	0.9660	0.9689	0.9666
24	0.9700	0.9720	0.9680	0.9667
25	0.9500	0.9540	0.9569	0.9550
26	0.9400	0.9540	0.9580	0.9555
27	0.9300	0.9370	0.9364	0.9354
28	0.9000	0.9380	0.9372	0.9351
29	0.9000	0.9080	0.9026	0.9034
30	0.8800	0.9160	0.9035	0.9028
31	0.8300	0.8690	0.8538	0.8537
32	0.8300	0.8670	0.8510	0.8534
33	0.7700	0.7880	0.7799	0.7803
34	0.7900	0.7790	0.7758	0.7737
35	0.7100	0.6630	0.6733	0.6764
36	0.6700	0.6320	0.6481	0.6497
37	0.5100	0.5130	0.5304	0.5371
38	0.4000	0.4190	0.4281	0.4307
39	0.4800	0.4190	0.4299	0.4293
40†	0.0000	0.0000	0.0000	0.0000

APPENDIX G
PROBABILITIES FOR THE BURSA NETWORK

j	$\Pi_\iota^{100}(v_j)$	$\Pi_\iota^{1000}(v_j)$	$\Pi_\iota^{10000}(v_j)$	$\Pi_\iota^{100000}(v_j)$
1	0.0100	0.0060	0.0055	0.0062
2	0.0500	0.0290	0.0330	0.0338
3	0.0300	0.0210	0.0163	0.0170
4	0.0000	0.0100	0.0067	0.0073
5	0.0400	0.0060	0.0055	0.0051
6	0.0000	0.0120	0.0092	0.0095
7	0.0000	0.0100	0.0095	0.0097
8†	0.0000	0.0000	0.0000	0.0000
9	0.0100	0.0100	0.0088	0.0086
10	0.0300	0.0300	0.0304	0.0319
11†	0.0000	0.0000	0.0000	0.0000
12†	0.0000	0.0000	0.0000	0.0000

APPENDIX H
PROBABILITIES FOR THE HANOI NETWORK

j	$\Pi_{\iota}^{100}(v_j)$	$\Pi_{\iota}^{1000}(v_j)$	$\Pi_{\iota}^{10000}(v_j)$	$\Pi_{\iota}^{100000}(v_j)$
1†	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000
4	0.0700	0.1090	0.1026	0.1029
5	0.1500	0.2160	0.2147	0.2148
6	0.2600	0.3460	0.3403	0.3427
7	0.3400	0.4690	0.4586	0.4620
8	0.4300	0.5460	0.5399	0.5427
9	0.5200	0.5950	0.5848	0.5858
10	0.5800	0.5970	0.5949	0.5934
11	0.7300	0.6880	0.7011	0.6932
12	0.8300	0.7600	0.7766	0.7717
13	0.8900	0.8280	0.8317	0.8317
14	0.6400	0.5830	0.5619	0.5595
15	0.5400	0.5300	0.4906	0.4894
16	0.4100	0.4050	0.3826	0.3779
17	0.2800	0.3150	0.3109	0.3061
18	0.2200	0.2090	0.2099	0.2054
19	0.1600	0.1060	0.1084	0.1037
20	0.0600	0.0550	0.0598	0.0625
21	0.0600	0.0610	0.0729	0.0735
22†	0.0000	0.0000	0.0000	0.0000
23	0.2400	0.2310	0.2467	0.2388
24	0.3600	0.3360	0.3452	0.3416
25	0.3300	0.3770	0.3823	0.3805
26	0.4500	0.4430	0.4322	0.4284
27	0.4600	0.4510	0.4384	0.4303
28	0.3600	0.3870	0.3756	0.3729
29	0.4600	0.4500	0.4475	0.4418
30	0.4900	0.4430	0.4503	0.4432
31	0.3300	0.3770	0.3823	0.3805
32	0.3300	0.3770	0.3823	0.3805

APPENDIX I
PROBABILITIES FOR THE KOBE NETWORK

j	$\Pi_{\iota}^{100}(v_j)$	$\Pi_{\iota}^{1000}(v_j)$	$\Pi_{\iota}^{10000}(v_j)$	$\Pi_{\iota}^{100000}(v_j)$
1†	0.0000	0.0000	0.0000	0.0000
2	0.2300	0.2890	0.2552	0.2565
3	0.2200	0.1600	0.1688	0.1655
4	0.1100	0.0440	0.0418	0.0415
5	0.1700	0.1460	0.1533	0.1464
6†	0.0000	0.0000	0.0000	0.0000
7	0.1700	0.1480	0.1490	0.1427
8	0.1800	0.1640	0.1653	0.1623
9	0.1200	0.1330	0.1400	0.1330
10	0.0700	0.0440	0.0509	0.0483
11	0.0600	0.0260	0.0268	0.0277
12	0.0600	0.0270	0.0288	0.0293
13	0.1100	0.0250	0.0338	0.0343
14	0.0000	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000

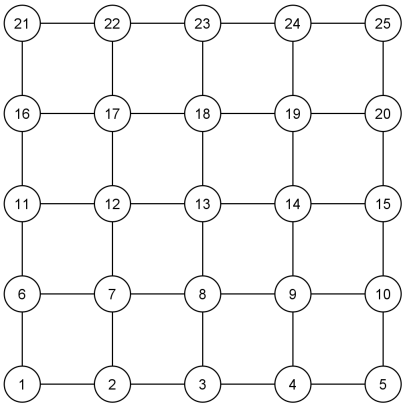


Fig. 7: Network 5.

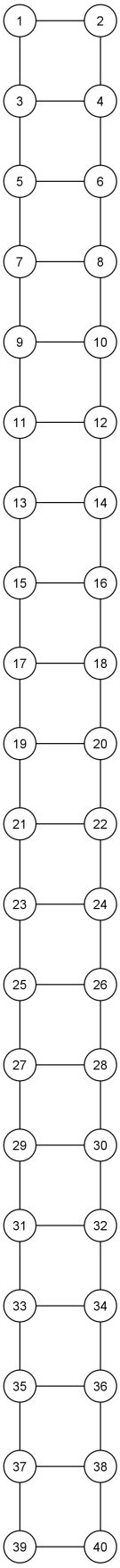


Fig. 8: Network 6.

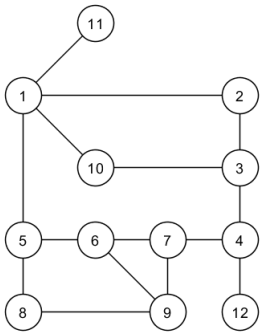


Fig. 9: Bursa network.

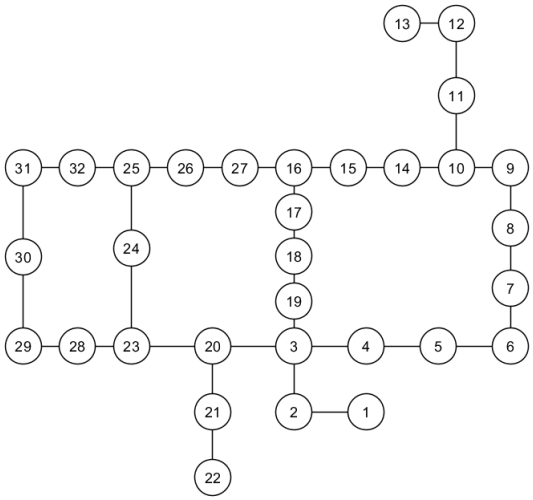


Fig. 10: Hanoi network.

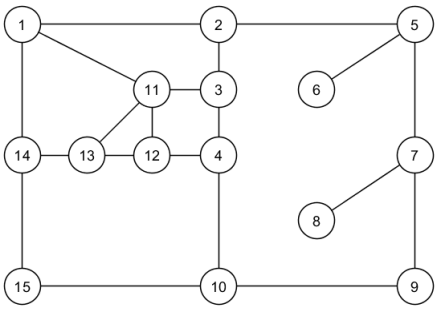


Fig. 11: Kobe network.

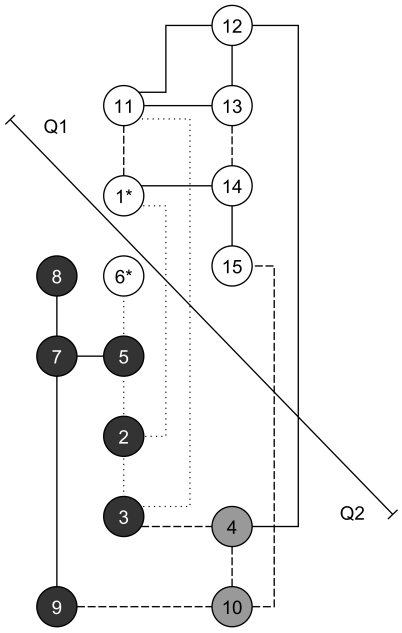


Fig. 12: The case of the Kobe water distribution network.

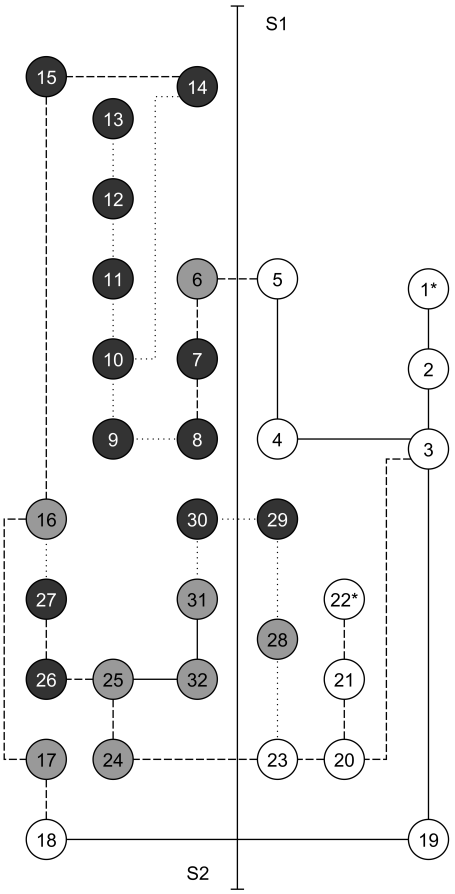


Fig. 13: The case of the Hanoi water distribution network.

TABLE I: Summary figures of the experiment on baselines with $N = 100$.

G	$\|E\|$	$\|V\|$	$t_{0.50,1}$	$t_{0.50,2}$	$t_{0.50}$	$t_{0.05,1}$	$t_{0.05,2}$	$t_{0.05}$	$t_{0.95,1}$	$t_{0.95,2}$	$t_{0.95}$	$\bar{t}$
1	15	6	0.0103	0.1736	0.1839	0.0024	0.1236	0.1261	0.0021	0.1919	0.1940	0.1680
2	30	20	0.0033	1.0636	1.0669	0.0162	0.5781	0.5943	0.0032	1.9426	1.9458	1.2023
3	23	14	0.0178	0.9262	0.9440	0.0028	0.8109	0.8137	0.0024	1.3442	1.3465	1.0347
4	30	20	0.0044	1.9757	1.9801	0.0038	1.2902	1.2940	0.0025	2.3726	2.3751	1.8831
5	40	25	0.0085	3.1381	3.1466	0.0036	2.4443	2.4478	0.0036	3.1186	3.1222	2.9056
6	58	40	0.0053	1.7131	1.7183	0.0033	1.0292	1.0325	0.0033	4.5098	4.5132	2.4213

TABLE II: Summary figures of the experiment on baselines with $N = 1000$.

G	$\|E\|$	$\|V\|$	$t_{0.50,1}$	$t_{0.50,2}$	$t_{0.50}$	$t_{0.05,1}$	$t_{0.05,2}$	$t_{0.05}$	$t_{0.95,1}$	$t_{0.95,2}$	$t_{0.95}$	$\bar{t}$
1	15	6	0.0053	1.7947	1.8001	0.0051	1.2479	1.2530	0.0059	2.1511	2.1570	1.7367
2	30	20	0.0105	9.9300	9.9405	0.0083	5.8300	5.8383	0.0090	17.5446	17.5536	11.1108
3	23	14	0.0085	11.0732	11.0817	0.0149	7.9114	7.9262	0.0068	12.3743	12.3811	10.4630
4	30	20	0.0100	16.5517	16.5617	0.0094	12.7018	12.7112	0.0100	22.5323	22.5423	17.2718
5	40	25	0.0211	32.8138	32.8349	0.0108	24.4238	24.4346	0.0106	45.9786	45.9892	34.4196
6	58	40	0.0164	18.2109	18.2273	0.0174	12.0532	12.0706	0.0171	68.3464	68.3635	32.8871

TABLE III: Summary figures of the experiment on baselines with $N = 10000$.

G	$\|E\|$	$\|V\|$	$t_{0.50,1}$	$t_{0.50,2}$	$t_{0.50}$	$t_{0.05,1}$	$t_{0.05,2}$	$t_{0.05}$	$t_{0.95,1}$	$t_{0.95,2}$	$t_{0.95}$	$\bar{t}$
1	15	6	0.0390	17.3316	17.3706	0.0376	12.3935	12.4311	0.0365	20.3875	20.4240	16.7419
2	30	20	0.0732	107.6360	107.7092	0.0689	55.6890	55.7579	0.0682	193.2180	193.2862	118.9178
3	23	14	0.0563	122.2990	122.3553	0.0636	84.4677	84.5313	0.0587	151.2630	151.3217	119.4028
4	30	20	0.0737	184.9180	184.9917	0.0744	140.5700	140.6444	0.0692	237.3070	237.3762	187.6708
5	40	25	0.0969	361.1590	361.2559	0.0929	252.3040	252.3969	0.0896	435.6780	435.7676	349.8068
6	58	40	0.1342	199.3880	199.5222	0.1306	116.1300	116.2606	0.1297	684.7590	684.8887	333.5572

TABLE IV: Summary figures of the experiment on baselines with $N = 100000$.

G	$\|E\|$	$\|V\|$	$t_{0.50,1}$	$t_{0.50,2}$	$t_{0.50}$	$t_{0.05,1}$	$t_{0.05,2}$	$t_{0.05}$	$t_{0.95,1}$	$t_{0.95,2}$	$t_{0.95}$	$\bar{t}$
1	15	6	0.3253	172.1320	172.4573	0.3180	109.7850	110.1030	0.3173	207.7550	208.0723	163.5442
2	30	20	0.6377	1022.9700	1023.6077	0.6194	521.9100	522.5294	0.6193	1836.6500	1837.2693	1127.8021
4	23	14	0.4895	1124.5700	1125.0595	0.4777	775.5610	776.0387	0.4874	1261.7900	1262.2774	1054.4586
5	30	20	0.6569	1674.9800	1675.6369	0.6277	1264.4100	1265.0377	0.6249	2215.3100	2215.9349	1718.8698
8	40	25	0.8540	3283.4000	3284.2540	0.8323	2463.4500	2464.2823	0.8323	4070.1600	4070.9923	3273.1762
9	58	40	1.2304	2237.0600	2238.2904	1.1988	1120.3600	1121.5588	1.1990	6780.1100	6781.3090	3380.3860

TABLE V: Summary figures of the experiment on applications with $N = 100$.

G	$\|E\|$	$\|V\|$	t_1	t_2	t
Bursa	15	12	0.0015	0.3540	0.3555
Hanoi	34	32	0.0033	1.4145	1.4178
Kobe	20	15	0.0025	0.5263	0.5288

TABLE VI: Summary figures of the experiment on applications with $N = 1000$.

G	$\|E\|$	$\|V\|$	t_1	t_2	t
Bursa	15	12	0.0042	3.6731	3.6773
Hanoi	34	32	0.0099	12.7125	12.7224
Kobe	20	15	0.0057	5.6930	5.6987

TABLE VII: Summary figures of the experiment on applications with $N = 10000$.

G	$\|E\|$	$\|V\|$	t_1	t_2	t
Bursa	15	12	0.0372	36.7711	36.8083
Hanoi	34	32	0.0985	114.5840	114.6825
Kobe	20	15	0.0505	57.1823	57.2328

TABLE VIII: Summary figures of the experiment on applications with $N = 100000$.

G	$\|E\|$	$\|V\|$	t_1	t_2	t
Bursa	15	12	0.3722	406.2250	406.5972
Hanoi	34	32	0.7168	1079.8100	1080.5268
Kobe	20	15	0.4222	543.7810	544.2032

REFERENCES

[1] W. Ellens and R. Kooij, "Graph measures and network robustness," *arXiv e-print 1311.5064*, 2013.

[2] B. Mojar, "The laplacian spectrum of graphs," *Graph Theory, Combinatorics, and Applications: Proceedings of the 6th Quadrennial International Conference on the Theory and Applications of Graphs Vol. 2*, pp. 871–898, 1991.

[3] H. Perez-Roses, "Sixty years of network reliability," *Mathematics in Computer Science Vol. 12*, pp. 275–293, 2018.

[4] R. Tan, F. Sikora, K. Ikeda, and K. See, *Transactions on Engineering Technologies*. Springer, 2021, ch. Arc Routing Based on the Zero-Suppressed Binary Decision Diagram, pp. 105–120.

[5] T. Toda, T. Saitoh, H. Iwashita, J. Kawahara, and S. Minato, "ZDDs and enumeration problems: State-of-the-art techniques and programming tool," *Computer Software Vol. 34 No. 3*, pp. 97–120, 2017.

[6] S. Minato, "Zero-suppressed BDDs for set manipulation in combinatorial problems," *Proceedings of the 30th International Design Automation Conference*, pp. 272–277, 1993.

[7] D. Knuth, *The Art of Computer Programming Vol. 4 Fasc. 1*. Addison-Wesley, 2009.

[8] S. Minato, "Zero-suppressed BDDs and their applications," *International Journal on Software Tools for Technology Transfer Vol. 3 No. 2*, pp. 156–170, 2001.

[9] R. Yoshinaka, J. Kawahara, S. Denzumi, H. Arimura, and S.-I. Minato, "Counterexamples to the long-standing conjecture on the complexity of BDD binary operations," *Information Processing Letters Vol. 112 No. 16*, pp. 636–640, 2012.

[10] H. Iwashita, Y. Nakazawa, J. Kawahara, T. Uno, and S. Minato, "ZDD-based computation of the number of paths in a graph," *Hokkaido University Division of Computer Science TCS Technical Report TCS-TR-A-12-60*, 2012.

[11] J. Kawahara, T. Inoue, H. Iwashita, and S. Minato, "Frontier-based search for enumerating all constrained subgraphs with compressed representation," *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences Vol. E100-A No. 9*, pp. 1773–1784, 2017.

[12] R. Tan, J. Kawahara, K. Ikeda, A. Garciano, and K. See, "Concerning a decision-diagram-based solution to the generalized directed rural postman problem," *IAENG International Journal of Computer Science Vol. 47 No. 2*, pp. 302–309, 2020.

[13] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, 1986.

[14] H. Iwashita and S. Minato, "Efficient top-down ZDD construction techniques using recursive specifications," *Hokkaido University Division of Computer Science TCS Technical Report TCS-TR-A-13-69*, 2013.

[15] R. Tan, J. Kawahara, A. Garciano, and I. Sin, "A zero-suppressed binary decision diagram approach for constrained path enumeration," *Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering 2019, WCE 2019, 3-5 July 2019, London, United Kingdom*, pp. 132–136, 2019.

[16] F.-M. Yeh and S.-Y. Kuo, "Obdd-based network reliability calculation," *Electronics Letters Vol. 33 No. 9*, pp. 759–760, 1997.

[17] S. Chaterjee, V. Ramana, G. Vishwakarma, and A. Verma, "An improved algorithm for k-terminal probabilistic network reliability analysis," *Journal of Reliability and Statistical Studies Vol. 10 Iss. 1*, pp. 15–26, 2017.

[18] M. Javanbarg, C. Scawthorn, J. Kiyono, and Y. Ono, "Reliability analysis of infrastructure and lifeline networks using obdd," *In Proceedings of the 10th International Conference on Structural Safety and Reliability*, pp. 3463–3470, 2010.

[19] A. Vasan and S. Simonovic, "Optimization of water distribution network design using differential evolution," *Journal of Water Resources Planning and Management Vol. 136 Iss. 2*, pp. 279–287, 2010.

[20] M. Javanbarg, J. Kiyono, and M. Ghazizadeh, "Reliability analysis of lifeline networks using binary decision diagram," *In Proceedings of the 4th International Conference on Modern Research in Civil Engineering, and Architectural and Urban Development*, 2016.

[21] D. Morrison, E. Sewell, and S. Jacobson, "Solving the pricing problem in a branch-and-price algorithm for graph coloring using zero-suppressed binary decision diagrams," *INFORMS Journal on Computing Vol. 28 No. 1*, pp. 67–82, 2016.

[22] A. Selcuk and M. Yucemen, "Reliability of lifeline networks with multiple sources under seismic hazard," *Natural Hazards Vol. 21*, pp. 1–18, 2000.

[23] M. Javanbarg and S. Takada, "Redundancy model for water supply systems under earthquake environments," *Proceedings of the 5th International Conference on Seismology and Earthquake Engineering*, p. 16094, 2007.