Data Article

Benchmark instances for road network repair and restoration problems in the context of disaster response operations

Luana Souza Almeida*, Revanth Kodali, Floris Goerlandt

Dalhousie University, Department of Industrial Engineering, Maritime Risk and Safety (MARS) Group, 5269 Morris Street, Halifax, Nova Scotia, Canada, B3H 4R2

A B S T R A C T

This article presents a database which contains a comprehensive and systematically varied set of network instances. These can be applied as benchmarks for multiple road repair and restoration problems in the context of natural disasters. The characteristics of the instances vary in terms of network size, intensity and type of disaster affecting the road network, the epicenter’s location, and the number of sub-networks in which the initial network is divided after the disaster occurs. The instances were developed primarily for the Multi-vehicle Prize Collecting Arc Routing for Connectivity Problem (KPC-ARCP). These are however easily adaptable to other well-known connectivity, vehicle routing, and facility location problems in the Operations Research literature. The instances are available on a public repository, as is the Python code to generate the instances.

© 2022 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

A R T I C L E   I N F O

Article history:
Received 29 June 2022
Revised 4 October 2022
Accepted 10 October 2022
Available online 14 October 2022

Dataset link: Benchmark instances for the multi-vehicle prize collecting arc routing for connectivity problem (Original data)

Keywords:
Natural disaster
Roads network
Road clearing
Prize collecting
Arc routing

* Corresponding author.
E-mail address: luana.almeida@dal.ca (L. Souza Almeida).
Social media: @Luana_S_Almeida, https://twitter.com/Luana_S_Almeida (L. Souza Almeida)

https://doi.org/10.1016/j.dib.2022.108674
2352-3409/© 2022 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
### Specifications Table

| Subject | Management Science and Operations Research |
|---------|-------------------------------------------|
| Specific subject area | Disaster management, network disruptions, roads network |
| Type of data | Text file with tables (.txt) Images (.png) |
| How the data were acquired | The code to generate the instances was developed on Python 3.8.5, and the file (“KPC-ARCP – Benchmarking Instances Generator.ipynb”) is available in Souza Almeida et al. [1]. The code automatically saves the instances files (text files and images), and it was executed in an AMD Ryzen 9 3960 × 16-core processor with 64 GB of RAM on June 13 to 24, 2022. |
| Data format | Raw Consolidated |
| Description of data collection | The instances were created following three steps. First, an initial network with all the edges cleared is created, considering selected pre-determined characteristics of the instance, such as its size. This initial network is called the “pre-disaster” network. The second step is to determine the edges to be blocked, which is done assuming that the disaster intensity follows a Bivariate Gaussian Distribution. This network with blocked edges is called the “post-disaster” network. Finally, in the third step, the graphs are plotted and a text file with the instance data is generated. |
| Data source location | - Institution: Dalhousie University - City/Town/Region: Halifax, Nova Scotia - Country: Canada |
| Data accessibility | The benchmarking instances and the Python code (.ipynb) are available in a public repository: Repository name: Mendeley Data Data identification number: 7mdfcs8f75 Direct URL to data: https://data.mendeley.com/datasets/7mdfcs8f75/ |

### Value of the Data

- The benchmarking instances introduced in this paper are useful for developing and testing network repair and restoration models, especially the “Multi-vehicle Prize Collecting Arc Routing for Connectivity Problem” (KPC-ARCP). The instances can be used by researchers in the field of Operations Research and Computer Science working with road network repair and restoration problems.
- As discussed in a recent review [2], researchers in the field have been developing and testing models on different instances. Therefore, it is not possible to compare the performance (i.e. objective function value and processing times) of the existing solution approaches. This set of instances is a starting point to unify the tests on KPC-ARCP as it allows researchers to test their approaches on the same instances.
- Researchers working with KPC-ARCP and other arc routing for connectivity problems can test their exact, heuristic and metaheuristic solution approaches on over 1,500 instances with systematically varied network structures. For instance, optimization models can be tested for medium and high intensity disasters and for different depot locations, among other characteristics. In addition, the instances can be a basis for developers creating exact models for other network repair and restoration problems in the context of natural disasters.
- The instances were developed primarily to advance research on the KPC-ARCP. However, they are easily adapted to other well-known connectivity problems, such as “Arc Routing for Connectivity Problem” (ARCP) [3,4], “Multi-vehicle Arc Routing for Connectivity Problem” (K-ARCP) [5], and “Prize Collecting Arc Routing for Connectivity Problem” (PC-ARCP)
In the cases of ARCP and PC-ARCP, researchers can use the instances directly. The difference between these two problems and KPC-ARCP is the number of teams available at the single depot and the calculation of the objective function, which are not determined in the instances. Testing a K-ARCP model would require a small adaptation. K-ARCP considers multiple depots, while the developed instances contain only one depot. Thus, the developer has to select one or more additional depots.

- The network instances can be used as a basis to other problems such as emergency facility location in the context of natural disasters, distribution of relief supplies, and other vehicle routing problems. The network structure (e.g. nodes coordinates, distances, and edges) can be combined with additional data. For example, consider that a distribution problem will be tested in one of the network instances. The communities' demands, capacities of the trucks and their depots are some of the data that are not provided in the current dataset, and would thus have to be added to such a problem. Note that researchers could test the distribution within each subnetwork (i.e. component) in KPC-ARCP or they could disregard the damaged edges (i.e. assume all edges are unblocked) and test the distribution within the whole network.

1. Data Description

The database consists of 1,710 benchmarking network instances with increasing sizes. Each network is composed of three images (".png" files) (e.g. "E-25-BR-M-5-D9BR – graph.png", "E-25-BR-M-5-D9BR – surface.png", and "E-25-BR-M-5-D9BR – damaged graph.png"), and one text file (e.g. "E-25-BR-M-5-D9BR.txt"). The text file is the main document, because it contains all the network's characteristics. These are: the name and type of the instance, the number of components, the graph average degree, the number and position of the nodes, the list of all edges, and the repair times of the damaged/blocked edges. In this paper, we use the terms blocked and damaged interchangeably, both meaning that a specific edge needs to cleared or fixed before it can be traversed. The network images can be reconstructed using only the data in the text file, but to facilitate the user's interpretation of the data, they were added to the database. Note that the images are mainly useful for smaller graphs, because larger networks contain so many edges and vertices that it is not possible to retrieve all relevant information about the network visually. For larger networks, the images mainly serve to illustrate the shape of the network and the damaged region.

The instances are named following the logic depicted in Fig. 1. The first letter ("E" or "H") indicates the type of disaster which is simulated to damage the network, with "E" for earthquake and "H" for hurricane. The first number specifies the number of nodes in the network. For instance, in "E-25-BR-M-5-D9BR", there are 25 nodes. The following letter indicates the location of the disaster's epicenter in the area covered by the network. This is indicated with "C" if it is situated in the center, with furthermore "UR" referring to the upper right corner, "UL" to the upper left corner, "BR" to the bottom right corner, and "BL" to the bottom left corner. The disaster intensity is specified with "M" for medium impact and "H" for high impact. When the disaster occurs, it is assumed that the initial network is subdivided into multiple disconnected subnetworks called "components", as defined in the previous studies of KPC-ARCP [6–8]. The second number in the instance name refers to the exact number of components of the damaged network. For example, in "E-25-BR-M-5-D9BR" there are five components. Finally, the depot ID and its position are specified. For example, "D9BR" means that node 9 is the teams' depot and is located at the bottom right corner. The exact meaning of the above introduced labels is explained in the Sections below.

The remainder of this Section is divided as follows. Section 1.1 explains the structure of the text file, and Section 1.2 demonstrates the content of the figures and how to interpret them.
1.1. Instance’s Text File

As mentioned above, each instance is characterized in four files: one text file and three images. In this Subsection, the structure of the text file is explained based on the example in Fig. 2. The file’s organization is analogous to the TSPLIB 95 introduced by Reinelt [9].

The first part of the file heading includes “NAME”, “TYPE”, and “COMMENT”. The “NAME” corresponds to the instance name, which notation is explained in Section 1. The “TYPE” specifies the type of problem that can be solved with this specific instance. This database was designed primarily for the “Multi-vehicle prize collecting arc routing problem” (KPC-ARCP) [6]. However, it can be easily used on other problems like ARCP, PC-ARCP, and K-ARCP. The developers of the instances are listed in the “COMMENT” section.

The second part of the heading explains the main metrics of the network. The number of nodes is described in “DIMENSION”. The length of the edges is the Euclidean distance between edge’s endpoints. Since the nodes are located on a 2-dimensional Euclidean grid, the “EDGE_WEIGHT_TYPE” is “EUC_2D”. Next, the ID of the node suggested to be the depot is specified in the “DEPOT_ID” field. The traversing speed of the road clearing teams is specified in “SPEED” (with unit kilometers per hour). Similar to previous authors who studied KPC-ARCP (e.g. Akbari and Salman [6] and Souza Almeida and Goerlandt [7]), the suggested traversing speed is 50 km/h. Note that this speed is only for traversing a road, and it is not the same as the clearing rate of the teams. Another network metric is the “AVERAGE_DEGREE”, which indicates the overall level of connectivity of the nodes in the network. For instance, in Fig. 2, the average degree is 2.32. So, each node is connected by an average of 2.32 edges. Considering that real-world road networks presented by Kasaei and Salman [4], Akbari and Salman [6] and Souza Almeida et al. [10] are reported to be sparse, these benchmarking instances are assumed to have an average degree lower than five. The “COMPONENTS” field identifies in how many sub-networks the graph is divided after the disaster.

The second part of the text file contains the data itself. The “NODE_COORD_SECTION” provides the information about the Node IDs, and their coordinates on the x1- and x2-axis, respectively. The axis system is shown in Fig. 10, with x1 the horizontal and x2 the vertical axis (with units in kilometers). For example, in Fig. 2, the node with ID 0 (zero) is located at the coordinate position [-44.04, 12.02]. Next, the “LIST_OF_ALL_EDGES_DATA_SECTION” is a list of all the edges in the graph, including both the traversable and the blocked edges. Finally, “DAMAGED_EDGES_DATA_SECTION” lists the blocked edges and their expected clearing time (with units
in hours). For instance, edge (2,18) is expected to be unblocked in 63.87 hours, as shown in Fig. 2.

1.2. Network Instances Illustrations

Each benchmarking instance is illustrated in three figures. The example instance “E-25-BR-M-5-D9BR” is illustrated in Figs 3–5. The network before the disaster occurs is stored in the file with name ending as “-graph.png”, as in Fig. 3. The nodes are numbered and represent communities and roads intersections. All the edges have the same color (green) because in the pre-disaster phase all roads are open and thus traversable.

When the disaster occurs, some of the roads are damaged and/or covered with debris. These are considered blocked roads, and thus not traversable before they are cleared/repaiored. The post-disaster scenario is illustrated in the file with name ending as “-damaged graph.png”, as
shown in an example in Fig. 4. The epicenter (i.e. the position where the damage intensity is highest) is represented by a red star, with the large blue node showing the teams’ depot, and the small black nodes representing the communities and roads intersections. The damaged edges are shown in red.

The disaster intensity applied to the network is represented by a probability density surface saved in the file with naming ending as “- surface.png”. For example, Fig. 5 represents the disaster intensity for instance “E-25-BR-M-5-D9BR”. The x1 and x2 axes form the coordinate system in which the nodes are located, as indicated above in Section 1.1, whereas the intensity axis refers to the probability of where the road network can be expected to be damaged. The surface’s color varies from red near the epicenter to green where the region is lightly impacted.
2. Experimental Design, Materials and Methods

The mechanism developed to generate the instances is explained in this Section. The process consists of three phases, as illustrated in Fig. 6. The inputs and outputs are listed in Fig. 6 and explained in Sections 2.1 and 2.2.

First, the connected pre-disaster network is created following the procedure in Section 2.1. Then, considering the type of disaster, the set of edges that are blocked in the post-disaster network, as well as their unblocking times, are determined as explained in Section 2.2. Finally, the network information is stored in the text file, with a structure already explained in...
Table 1
Parameters used to construct the pre-disaster network.

| Notation | Parameter | Description                                    | Value          |
|----------|-----------|-----------------------------------------------|----------------|
| n        | Dimension | The number of nodes in the instance.          | 25, 50, 75, 100, 250, 400, 500, 750, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 8000, 10000 |
| [-x1, +x1], [-x2, +x2] | Grid size | Limits of the grid where the nodes are located. | [-100, 100], [-100, 100] |
| Δmax    | Upper bound node degree | Maximum number of edges connected to each vertex. | 5              |

Section 1.1. The Python code to generate the instances was written on Jupyter Notebooks and is available on Mendeley with the filename “KPC-ARCP – Benchmarking Instances Generator.ipynb”. The code was written using Python 3.9.7, and the libraries used were NetworkX 2.6.3, Scipy 1.7.1, Seaborn 0.11.2, Matplotlib 3.4.3, and Numpy 1.20.3. The computational power was an AMD Ryzen 9 3960 × 16-core processor with 64 GB of RAM.

2.1. Mechanism to Generate the Pre-Disaster Network

Consider the graph \( G = (V, E) \) where \( V \) is the set of nodes and \( E \) is the set of edges. The nodes correspond to the communities, roads’ intersections, and teams’ depot. The edges represent the roads. The network is assumed to be connected, with all the roads operable under normal conditions, see Fig. 3 for an example. This Section explains the procedure to create the pre-disaster network, following an approach inspired by Souza Almeida et al. [8], which is summarized in Fig. 7.

The input parameters and their respective values are listed in Table 1. The parameters are the number of nodes (i.e. network dimension), the size of the grid where the vertices are going to be located, and the maximum number of connections that each vertex can have (i.e. upper bound vertices degree).

Consider a graph with \( n \) nodes. An array with \( n \) entries is randomly generated considering a normal distribution between one and five (i.e. the “upper bound node degree”). The entries represent the expected degree of each node. For example, if node with ID zero has a degree of three, then node zero is supposed to be connected with three other nodes. After determining the nodes’ degrees, a NetworkX’s function called “expected_degree_graph” is applied.

The array with the vertices degrees does not guarantee a connected graph. Thus, the algorithm checks the network connectivity. If the pre-disaster network has multiple components, these need to be connected. The algorithm iterates over each component and links it with a single edge to the first component. The connecting edge is determined by the first node of the component and a randomly selected node in the first component. Fig. 8 exemplifies this procedure. There are three components \( (0, 1, \text{and } 2) \), and the initial network is disconnected. In iteration 1, node A, the first node of component 1, is linked to a randomly selected node of component 0 (node 2). Then, on the second iteration, node 4 (the first node of component 2) is linked to node 1 - a randomly selected node of component 0. Thus, after adding edges \( (A,2) \) and \( (4,1) \), the network becomes connected.

An observation related to the connectivity of the pre-disaster network must be addressed. If there are more than five components, the nodes in the components are updated on every iteration multiple of five. The update is performed to avoid linking all the disconnected components to the same nodes. For example, if component 0 had only node 0, then all the components would be connected to node zero, and consequently its degree would be higher than the upper bound node degree, which is five as indicated in Table 1.

After creating a connected graph, the adjacency matrix of the graph is retrieved. In addition, the nodes are positioned on the grid following the Fruchtman-Reingold layout. This layout
Fig. 7. Steps to generate a pre-disaster network.
positions the vertices so that the number of edges crossing each other is minimized, while trying to keep the edge's length similar [11]. Thereafter, the nodes' coordinates are collected.

A final step in creating the pre-disaster network is the creation of a list of candidate depots. To avoid creating disrupted networks with unconnected depots, a node can be a depot if it has a degree greater than the network's average degree. For example, if the network's average degree is 2.44, the only nodes with more than or equal three edges are added to the list. Then, the depot is randomly selected from this list, and its position in the grid is assessed and classified using labels “C”, “UL”, “UR”, “BL”, or “BR”. Finally, the length of each edge is calculated using the Euclidean distance.

### 2.2. Mechanism to Determine the Post-Disaster Network

After the disaster strikes, some of the roads are taken assumed to be damaged. Taking the pre-disaster network constructed as explained in Subsection 2.2 as a basis, this Subsection describes the procedure to select the edges that will be blocked by the disaster, and their clearing times. The procedure is summarized in Fig. 9.

The first step is to define the epicenter of the disaster. For earthquakes, we assume an idealized point location as the epicenter, acknowledging that real-world earthquakes may occur.
Fig. 9. Steps to determine the post-disaster network.

1. Determine the epicenter of the disaster
2. Calculate the Bivariate Gaussian in the midpoint of each edge
3. Remove all edges on region $\alpha$
4. Remove batches of edges located on region $\beta$
5. Create text file and plot network
Table 2
Intervals for axes x1 and x2 applied in the Uniform Distribution for determining the epicenter’s coordinates.

| Location | Interval - x1-axis | Interval - x2-axis |
|----------|--------------------|--------------------|
| C        | [0,0]              | [0,0]              |
| UL       | [-100,0]           | [0,100]            |
| UR       | [0,100]            | [0,100]            |
| BL       | [-100,0]           | [-100,0]           |
| BR       | [0,100]            | [-100,0]           |

According to different fault line shapes [12]. For hurricanes, we also apply the term “epicenter” for consistency in the dataset. Similarly, a point location is taken as an idealized center of the disaster intensity, which can be understood to reflect the eye of the hurricane, or the area where the damage is most intense. As for earthquakes, this center point determines the intensity of the damages through a Bivariate Gaussian Distribution, as explained in more detail below.

As described in Table 1, the grid ranges from -100 to +100 on each axis, see also Fig. 10. The expected region of the epicenter (i.e. “C”, “UL”, “UR”, “BR”, “BL”) is selected at the outset of the process of Fig. 9. Assume that “BR” is the region, as exemplified in Fig. 10. The coordinate on the x1-axis is randomly generated using a uniform distribution with interval [0, 100]. Similarly, the coordinate on the x2-axis is created with interval of [-100, 0]. Table 2 indicates the intervals used to determine the epicenters based on their location for all other location labels. As depicted in Fig. 10, the coordinates of the epicenter are (μ₁, μ₂).

The Bivariate Gaussian Distribution is used to determine the intensity of the disaster across the network. This distribution adopts a random vector \( Y = (Y_1, Y_2) \), where each random variable follows a normal distribution with their own mean and variance, such that \( Y_1 \sim N(\mu_1, \sigma_1^2) \) and \( Y_2 \sim N(\mu_2, \sigma_2^2) \). Note that the difference between how earthquakes and hurricanes are conceptualized when inflicting damage to the road networks in the instances, consists of the value of the parameters.

The mean vector (Eq. 01) has two elements: \( \mu_1 \) and \( \mu_2 \), which are the coordinates of the epicenter on the x1 and x2-axis, respectively.

\[
\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}
\]
The covariance matrix (Eq. 02) has four elements: the standard deviation $\sigma_1$ of the disaster area on the x1-axis, the standard deviation $\sigma_2$ of the disaster area on the x2-axis, and their respective variance. An important parameter is $\rho$, which determines the shape of the surface, which is taken as the feature distinguishing the disaster type.

$$\sum = \begin{bmatrix} \sigma_1^2 & \rho \sigma_1 \sigma_2 \\ \rho \sigma_1 \sigma_2 & \sigma_2^2 \end{bmatrix}$$

(2)

Consider the contours of the Gaussian Bivariate distribution in Fig. 11. The contour colors indicate the intensity of the disaster, with red most impact and green least. The contours also provide insight in the size of the region affected. Note that when $\rho = 0$, the affected region had the shape like a circle, whereas when $\rho = +0.8$ or $\rho = -0.8$, the region had the shape of an ellipse, as shown in Fig. 11. Considering that earthquakes are broader and that hurricanes normally move across a region in a more line-like movement pattern, it was decided to adopt $\rho = 0$ to represent damage patterns for earthquakes, and $\rho = -0.8$ and $+0.8$ to represent hurricane damage patterns.

The probability density function $f(x)$ of the Bivariate Gaussian Function at the point $x$ is calculated using (Eq. 03), where $\mu$ and $\sum$ are the matrices defined in (Eq. 01) and (Eq. 02).

$$f(x) = \frac{1}{\sqrt{(2\pi)^k \det \sum}} \exp \left( -\frac{1}{2} (x - \mu)^T \sum^{-1} (x - \mu) \right)$$

(3)

In the instances created in this dataset, the probability of the edges being blocked is related to its proximity to the epicenter, with the expected intensity calculated using (Eq. 03). Note that $x$ is a point, but the roads are represented as edges (lines). Thus, the intensity of the edge is assumed to be the probability calculated in its midpoint. For example, if the intensity of the midpoint of edge [1,2] is 0.06, then edge (1,2) has intensity of 0.06. The relevant parameters the mechanism to determine what roads in the network are blocked, are specified in Table 3.

Consider the schematic in Fig. 12. The epicenter has the highest disaster intensity, denoted as $i_{max}$ . Region $\alpha$ contains all the edges’ midpoints of which the disaster intensity $f(x)$ (calculated with Eq. 03), meets the criterion $0.7i_{max} \leq f(x) \leq i_{max}$ for the medium impact case, and $0.8i_{max} \leq f(x) \leq i_{max}$ for the high impact case, respectively. Region $\beta$ contains all the edges’ midpoints of which the disaster intensity $f(x)$ meets the criterion $0.3i_{max} \leq f(x) \leq i_{max}$ and $0.2i_{max} \leq f(x) \leq i_{max}$ for the medium and high impact case, respectively.

After identifying the edges located in regions $\alpha$ and $\beta$, the algorithm starts to identify the roads to be blocked. When a road is identified to be blocked, the associated clearing time is randomly generated using a uniform distribution between 0 and 72 hours.

The edges are blocked in three steps, as shown in Fig. 13. First, all the edges located in the $\alpha$ region are blocked, and the number of components is checked. If the network has more components than a pre-determined number of components (see Table 4), one edge at a time is randomly reconnected, until the expected number of components as specified in the instance definition is met. In some iterations, it is possible that after blocking all the edges in the alpha region, the number of components is less than expected. In such cases, a batch of $b$ random edges is blocked, and the number of components is checked. In this paper, it was adopted a batch of size five. This process is repeated until the exact number of components is met.

### Table 3
Parameters adopted in the Bivariate Gaussian Distribution.

| Parameter | Medium impact | High impact |
|-----------|---------------|-------------|
| $\mu_1, \mu_2$ | C, UL, UR, BL, BR | C, UL, UR, BL, BR |
| $\rho$ | Hurricanes: -0.8 and 0.8 | Hurricanes: -0.8 and 0.8 |
| $\sigma_1$ | 30 | 30 |
| $\sigma_2$ | 40 | 40 |
Fig. 11. Relationship between the type of disaster and the parameter \( \rho \).

- \( \rho = 0 \) (Earthquake)
- \( \rho = -0.8 \) (Hurricane)
- \( \rho = 0.8 \) (Hurricane)
L. Souza Almeida, R. Kodali and F. Goerlandt / Data in Brief 45 (2022) 108674

Fig. 12. Upper and lower bound limits of the disaster's intensity.

Table 4
Expected number of components per instance size.

| Instance Size | Case 1 | Case 2 | Case 3 |
|---------------|--------|--------|--------|
| 25            | 3      | 5      | 7      |
| 50            | 4      | 7      | 10     |
| 75            | 4      | 7      | 10     |
| 100           | 5      | 9      | 15     |
| 250           | 5      | 9      | 49     |
| 400           | 6      | 11     | 60     |
| 500           | 7      | 12     | 87     |
| 750           | 8      | 13     | 138    |
| 1000          | 9      | 14     | 163    |
| 1500          | 9      | 14     | 211    |
| 2000          | 10     | 15     | 219    |
| 2500          | 10     | 15     | 365    |
| 3000          | 11     | 16     | 571    |
| 3500          | 12     | 17     | 576    |
| 4000          | 13     | 18     | 687    |
| 4500          | 13     | 18     | 871    |
| 5000          | 14     | 19     | 910    |
| 8000          | 14     | 19     | 1302   |
| 10000         | 15     | 25     | 1947   |

the network achieves the target number of components, the algorithm is stopped. Then, the text file is created and the three images of the network are plotted.

In this paper, the instances are created considering three cases of number of components, with details shown in Table 4. Cases 1 and 2 are intended to produce networks with few components, whereas case 3 aims to divide the initial network in many parts.

As described in Fig. 1, each instance has six main characteristics: disaster type, instance size, epicenter location, disaster intensity, number of components, and depot location. Since there are two disasters types with in total three values for the shape of the impacted area ($\rho = -0.8$, $\rho = 0$, $\rho = +0.8$), nineteen sizes of the instances, five epicenter's locations, two intensities, and three options for the number of components (cases 1, 2, and 3), and one depot, there are in total 1,710 benchmarking instances available in the database.
Fig. 13. Logic block edges considering their position on regions α and β.
Ethics Statements

This work meets the requirements of ethics as stated in (https://www.elsevier.com/journals/data-in-brief/2352-3409/guide-for-authors) and (https://www.elsevier.com/about/policies/publishing-ethics#Authors). This work also does not involve studies with animals and humans.

CRediT Author Statement

Luana Souza Almeida: Conceptualization, Methodology, Software, Data Curation, Writing – original draft, Visualization; Revanth Kodali: Methodology, Software, Visualization; Floris Goerlandt: Conceptualization, Methodology, Writing – review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Benchmark instances for the multi-vehicle prize collecting arc routing for connectivity problem (Original data) (Mendeley Data).

Acknowledgments

This work was supported by the Province of British Columbia and the Marine Environmental Observation Prediction and Response [grant numbers 2-02-03-040.2, 2-02-03-041.3]. The last author is furthermore supported by the Canada Research Chairs Program, through a grant by the Natural Sciences and Engineering Research Council (NSERC). This financial support is gratefully acknowledged.

References

[1] L. Souza Almeida, R. Kodali, F. Goerlandt, Benchmark instances for the multi-vehicle prize collecting arc routing for connectivity problem, Mendeley Data (1) (2022), doi:10.17632/7mdfc8s875.1.
[2] L. Souza Almeida, F. Goerlandt, R. Pelot, Trends and gaps in the literature of road network repair and restoration in the context of disaster response operations, Socioeco. Plann. Sci. (101398) (2022), doi:10.1016/j.seps.2022.101398.
[3] A.N. Asaly, F.S. Salman, Arc selection and routing for restoration of network connectivity after a disaster, in: B.Y. Kara, I. Sabuncuoglu, B. Bidanda (Eds.), Global Logistics Management, Taylor & Francis Group, London, 2014, p. 316.
[4] M. Kasaei, F.S. Salman, Arc routing problems to restore connectivity of a road network, Transp. Res. Part E. 95 (2016) 177–206, doi:10.1016/j.tre.2016.09.012.
[5] V. Akbari, F.S. Salman, Multi-vehicle synchronizined arc routing problem to restore post-disaster network connectivity, Eur. J. Oper. Res. 257 (2) (2017) 625–640, doi:10.1016/j.ejor.2016.07.043.
[6] V. Akbari, F.S. Salman, Multi-vehicle prize collecting arc routing for connectivity problem, Comput. Oper. Res. 82 (2017) 52–68, doi:10.1016/j.cor.2017.01.007.
[7] L. Souza Almeida, F. Goerlandt, An ant colony optimization approach to the multi-vehicle prize-collecting arc routing for connectivity problem, Multimodal Transp. 1 (3) (2022), doi:10.1016/j.multra.2022.100033.
[8] L. Souza Almeida, F. Goerlandt, R. Pelot, K. Sörensen, A Greedy Randomized Adaptive Search Procedure (GRASP) to the multi-vehicle prize collecting arc routing for connectivity problem, Comput. Oper. Res. 143 (105804) (Jul. 2022), doi:10.1016/j.cor.2022.105804.
[9] G. Reinelt, TSPLIB 95, Universität Heidelberg, Germany, 2013 Accessed: Apr. 25, 2022. [Online]. Available: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp95.pdf.
[10] L. Souza Almeida, et al., Datasets of disrupted transportation networks on Canada’s West Coast in a plausible M9.0 Cascadia Subduction Zone earthquake scenario, Data Brief (2022) Submitted.

[11] T.M.J. Fruchterman, E.M. Reingold, Graph drawing by force-directed placement, Softw. Pract. Exp. 21 (11) (Nov. 1991) 1129–1164, doi:10.1002/spe.4380211102.

[12] B.F. Atwater, S. Musumi-Rokkaku, K. Satake, Y. Tsuji, K. Ueda, D.K. Yamaguchi, The Orphan Tsunami of 1700 : Japanese Clues to a Parent Earthquake in North America, University of Washington Press, Reston, Va, 2015 vol. Second edition = 第2版, no. Vol. 1707[Online]. Available: http://ezproxy.library.dal.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xna&AN=1226453&site=ehost-live.