**Supplementary Note 1**

***Segment Drop-Link Criticality***

This metric quantifies the impact (‘cost’) of reducing or disabling a segment between all origin and destination pairs. It should be noted that these pairs can be defined as one-to-one, one-to-many, many-to-one and/or many-to-many based upon the network, origins and destinations being considered.

The matrix of optimal costs in the reference network is defined as $R$, and the matrix of optimal costs in the perturbed network $k$is defined as $P\_{k}$. The number of origins and targets specified by the user in the system are given respectively by $o$ and $t$. These matrices are computed using the Dijkstra cost algorithm (Dijkstra 1959) for every origin-destination pair:

$$R=(c\_{ij})\in R^{o×t}$$

$$P\_{k}=(c\_{ij})\_{k}\in R^{o×t}$$

The optimal cost matrices can be weighted to account for the relative higher importance of certain origins or destinations. For example, in an electricity network, the disruption of the connectivity of a power plant with an output of 1MW will have less impact on the grid than that of a 100MW plant. The origin weights diagonal matrix $W\_{o }$ and the target weights diagonal matrix $W\_{t}$ are defined as:

$$W\_{o}=(w\_{i}δ\_{ij})\in R^{o×o}$$

$$W\_{t}=(w\_{j}δ\_{ij})\in R^{t×t}$$

where $δ\_{ij}$is the Kronecker delta.

The weighted reference ($R^{w}$) and weighted perturbed ($P\_{k}^{w}$) matrices of optimal costs are consequently computed to be:

$$R^{w}=W\_{o}\*R\*W\_{t}$$

$$P\_{k}^{w}=W\_{o}\*P\_{k}\*W\_{t}$$

The impact ($I\_{k}$) of the removal of segment (or set of segments) $k$ on the network relative to the defined origins-destinations pairs is calculated using the Frobenius norm of the difference between the weighted perturbed ($P\_{j}^{w}$) and the weighted reference ($R^{w}$) matrices of costs.

$$I\_{k} = ||P\_{k}^{w}-R^{w}||\_{F}$$

The impacts $I\_{k}$ are then ranked. The higher the value, the more critical the corresponding segment. Due to its ranking nature, this metric is not applied to the Monte Carlo scenario.

***Computational performance***

The single-source, shortest-path algorithm from the BOOST C++ library is used through the pgRouting extension to PostgreSQL (pgRouting, 2021). This algorithm uses a Fibonacci heap min-priority queue, hence giving a time complexity of O(E + VlogV), where E is the number of edges and V the number of nodes (Fredman & Tarjan, 1987). Consequently, the time complexity of our implementation is at worst O(P \* S \* (E + VlogV)), where P is the number of perturbations considered and S is the number of origins fed to the single-source shortest-path algorithm.

To illustrate the actual performance one can expect on an average machine, Table S1 gives the average computation time per network perturbation according to the given number of origins and targets on the Dominica network (860 km, divided in segments of 100 meters with remainder lengths). One needs to multiply by the number of perturbations to obtain the expected total runtime. The computation is memory heavy, and with an 8GB system, one can expect to perform analyses for a system of up to around 1,000,000 origins-destinations pairs. This indicates a computational time of approximately 7,800 milliseconds (<1 minute).

|  |  |  |  |
| --- | --- | --- | --- |
| Number of O-D pairs | Number of Origins | Number of Destinations | Computation Time (ms) |
| 100 | 1 | 100 | 20 |
| 1,000 | 1 | 1000 | 22 |
| 5,000 | 1 | 5000 | 35 |
| 10,000 | 100 | 100 | 165 |
| 100,000 | 100 | 1000 | 692 |
| 500,000 | 100 | 5000 | 3302 |
| 1,000,000 | 1000 | 1000 | 7655 |

**Table S1.** Computation time, in ms, per perturbation solution over the Dominica road network, for varying number of O-D pairs