Mapping Network States using Connectivity Queries

Alexander Rodríguez*, Bijaya Adhikari†, Andrés D. González°, Charles Nicholson°,
Anil Vullikanti†† and B. Aditya Prakash*

*College of Computing, Georgia Institute of Technology [{arodriguezc, badityap}@cc.gatech.edu]

†Department of Computer Science, University of Iowa [bijaya-adhikari@uiowa.edu]

°School of Industrial and Systems Engineering, University of Oklahoma [{andres.gonzalez, cnicholson}@ou.edu]

††Biocomplexity Institute and Department of Computer Science, University of Virginia [vsakumar@virginia.edu]

***Abstract*—Can we infer all the failed components of an infrastructure network, given a sample of reachable nodes from supply nodes? One of the most critical post-disruption processes after a natural disaster is to quickly determine the damage or failure states of critical infrastructure components. However, this is non-trivial, considering that often only a fraction of components may be accessible or observable after a disruptive event. Past work has looked into inferring failed components given point probes, i.e. with a direct sample of failed components. In contrast, we study the harder problem of inferring failed components given partial information of some 'serviceable' reachable nodes and a small sample of point probes, being the first often more practical to obtain. We formulate this novel problem using the Minimum Description Length (MDL) principle, and then present a greedy algorithm that minimizes MDL cost effectively. We evaluate our algorithm on domain-expert simulations of real networks in the aftermath of an earthquake. Our algorithm successfully identifies failed components, especially the critical ones affecting the overall system performance.**

***Index Terms*—Network inference, data mining, critical infrastructure networks**

I. Introduction

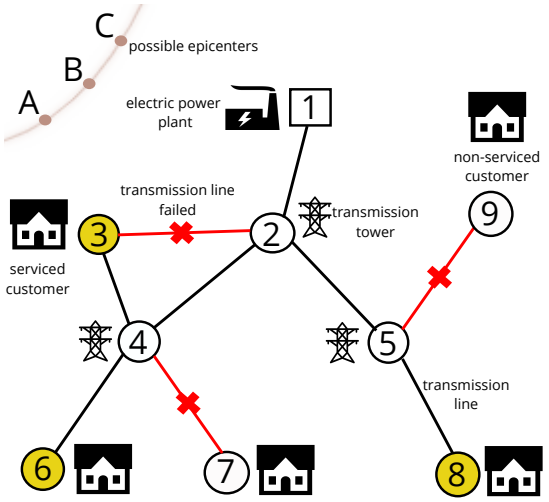
Motivation. Inferring network states from partial observations is an important problem in many applications, such as in epidemics, social networks and Internet/AS graphs. Consider critical infrastructure networks like water distribution system or electric grids — their rapid functional restoration is a priority right after a disaster such as an earthquake [1]. The sparse installation of real time monitoring systems installed in these networks (e.g. SCADA systems) imply that only a partial observation of the network is available post disaster i.e., we only get a sample of the functional states of the components ('failed' or not). However, for decision makers to devise an effective recovery plan, the information on functional states of the entire network is required. Hence, it is crucial to infer the states of the components in such networks after a disaster.

Despite its importance, there has been limited research in this space. Most past work has only looked into the so-called *point queries*, i.e. one is given some direct sample of failed components. In epidemics, Shah et al. [2] proposed inferring missing infections and source detection given a sample of infections. Indeed, in network tomography, all prior work has focused on independent failures [3]. Recently, there has been some progress on reconstructing network states assuming that the failures are geographically correlated given a sample of the failed components [4]. As mentioned earlier, a common theme among these works is that the observed failures are a direct sample of failed components.

A different class of queries, namely, *connectivity queries* also occur naturally. For example, consider an electric power grid, where transmission line connecting two component may fail [5]. Such failures cause disconnections in the network. SCADA systems (which oversee sensors in a power grid) provide information on whether or not there is a power supply at a specific demand node (a consumer node). These demand nodes are supplied by identifiable supply nodes (power generators), thus, a stop of supply to a demand node can be mapped to disconnectivity between it and the supply nodes. Similarly, the continuous supply on a demand node indicates that there is at least one connected path to the supply nodes. A motivating example is depicted in Figure 1. Here, node 1 is a source node (power plant), the set $\{3, 4, 6, 7, 8, 9\}$ is the set of demand nodes (consumers), and the remaining nodes $\{2, 4, 5\}$ are transmission nodes. If due to a disaster edges in the set $\{(2, 3), (4, 7), (5, 9)\}$ fail, this would disconnect nodes 7 and 9.

Why study connectivity queries? After a natural disaster, connectivity queries are *more practical* to obtain than point queries. Information about supplied demand and customer service (a natural source of connectivity queries) is often relatively accessible—customers nowadays have multiple ways of reporting issues, including even social media [6]—and has been proven to be useful to localize failures [7]. In our example, a sample of the nodes connected to the source are observed i.e, we observe a subset of $\{3, 6, 8\}$ thanks to signals from sensors and/or consumer reports.

Motivated by the reasons above, we investigate the usefulness of adding connectivity queries to the point queries (collectively referred to as 'joint queries') in inferring network failures. We find that using *only* connectivity queries for inference is very challenging. For instance, in our example, no demand node is affected by the failure of edge $\{2, 3\}$, which makes it undiscoverable to connectivity queries. Hence, we propose using 'joint queries' where we rely primarily on connectivity queries but also use a small sample of point queries to boost performance. To this end, we propose to the leverage the powerful Minimum Description Length (MDL) [8] concept, to find out both the number of failures and their identity. MDL is especially suited for this task, as it gives us a principled way to formulate our problem.



(a)

Notation	Physical set in Fig. 1(a)
Supply nodes V_S	Electric power plant $= \{1\}$
Transshipment nodes V_T	Transmission towers $= \{2, 4, 5\}$
Demand nodes V_D	Houses/customers $= \{3, 6, 7, 8, 9\}$
Serviced nodes $\mathcal{S}$	Houses with electricity $= \{3, 6, 8\} \subseteq V_D$
True failure set I	Transmission lines failed $= \{(2, 3), (4, 7), (5, 9)\}$
Disaster scenarios $\mathcal{O}$	Possible epicenters $= \{A, B, C\}$

(b)

Fig. 1. (a) An example of our problem using an electric power network disrupted by an earthquake. We refer to the demand nodes (houses) still connected to a supply node (power plant) as serviced nodes, i.e. houses with electricity. Thus, given a sample of serviced nodes $\{3, 8\}$ (connectivity probes), a smaller sample of the set of failed edges I, and knowing that the epicenter can be A, B, or C, we want to infer the rest of true failure set (edges with crosses). Note that failed edge $\{2, 3\}$ does not affect the serviceability of any node; therefore, using only connectivity probes will not allow us to discover this failure. (b) Description of the physical sets in (a), and its association with our notation.

Contributions. Our main contributions in this paper are as follows:

1) *Formulation and algorithm.* We formulate NETPATHSTATE to leverage joint probes for network state inference. We propose JOINTPATHMAP algorithm to solve the problem.
2) *Extensive experiments on real datasets*. We use domain-expert simulations by civil engineers of three real network topologies in the aftermath of an earthquake. Our extensive experiments demonstrate that JOINTPATHMAP significantly outperforms the baselines.

To the best of our knowledge, our work is the first to propose connectivity queries and joint queries for network state inference.

II. PRELIMINARIES

A. Context and Setup

We are given an undirected graph $G(V, E)$ that represents an infrastructure network. Links in E represent transmission systems and set V contains three disjoint sets of nodes: supply nodes V_S, demand nodes V_D, and transshipment nodes V_T. Thus, $V = V_S \cup V_D \cup V_T$. Supply nodes are where flow (e.g. electricity, water) is originated, demand nodes are where flow is received and distributed to final users, and transshipment nodes are where flow is transferred to other link(s). We assume supply and demand nodes are related by many-to-many relationships (i.e. one supply node can service many demand nodes and one demand node can be serviced by many supply nodes) as it is observed in electric and water networks. This kind of structure appears in many different critical infrastructure networks [5].

In addition, we are given a set $\mathcal{O}$ containing disaster scenarios that characterize the most likely disaster hazards. For example, in case of earthquakes, they can be described by a magnitude and a geographical epicenter; even though there is a myriad of possibilities for this, typically there is a finite set of earthquake scenarios (our set $\mathcal{O}$) that can characterize most of the seismic hazard [9]. Once a scenario is characterized, the effects on the infrastructure network are driven by soil and structural properties in a stochastic manner.

B. Failure Model

The failure model leverages 'geographically correlated' failures, which have been extensively used in prior work both in the computer science [10] and civil engineering communities [1]. At an abstract level, given a disaster scenario $o \in \mathcal{O}$, some set of edges $I \subseteq E$ fail. Let $p(o)$ be the probability of a disaster scenario o to had happen; hence, by previous definition, $\sum_{o \in \mathcal{O}} p(o) = 1$. Let $F(e|o)$ be the probability of edge e to fail given o, e.g. probability of an electric transmission line to fail given an earthquake with epicenter o. These probability distributions are calculated considering not only the distance to the epicenter of o, but also the type of structure, construction materials, and condition. In addition, we assume all edge failures depend only on the disaster's effect, i.e. failures are independent from each other. Both $\mathcal{O}$ and $F(e|o), \forall e \in E, \forall o \in \mathcal{O}$ are given by civil engineering guidelines, as in our experiments.

III. FORMULATION AND APPROACH

Here we systematically formulate our problem based on network connectivity using the MDL principle. See Table I for notation.

A. Connectivity Probes

Once a failure has taken place in the network, an operator performs the so-called "connectivity queries" on the network, as described next, to gather some information on the status. Connectivity queries (coming from sensors and/or customer reports) serve as indirect observers for the network states. As exemplified in Figure 1, when failures happen, they directly effect the *serviceability* of some demand nodes in the network. Demand nodes require an active path to a nearby supply node to be serviced (e.g. receive electricity). Thus, due to failures because of the disaster, some demand nodes become non-serviced. We define a 'sensor' set $\mathcal{S} \subseteq V_D$ as the set of demand nodes which are still serviced after removing failure edge set I from G. The operator will finally receive a *connectivity probe* set $\mathcal{Q}_c \subseteq \mathcal{S}$. We can assume that it is chosen uniformly at random (with some given probability, say γ_c), from $\mathcal{S}$. Therefore, we have the following dependency relationships:

$$o \rightarrow I \rightarrow \mathcal{S} \rightarrow \mathcal{Q}_c \tag{1}$$

B. Joint Probes

While connectivity queries are more practical to obtain, they are often not enough. If the demand nodes have multiple paths to the supply nodes (i.e., the network is redundant) the connectivity probes Q_c fail to provide enough information to infer I and o correctly. We saw this in Figure 1, where failure of edge $\{2, 3\}$ was not disconnecting any node, thus, it does not affect serviceability. It turns out that such phenomenon is quite common in real critical infrastructure networks as we empirically demonstrate it in Section IV. As the failures and serviceability are not interdependent in such networks, the connectivity probes do not relay any information regarding the failures; making it near impossible to infer the failures given the connectivity probes. The problem is compounded by the fact that the number of failure sets which do not effect the serviceability is exponential in terms of number of such edges. As we show in our experiments, using only connectivity probes returns failure set I where recall is low as it fails to identify the edges which have no effect on the serviceability.

Hence, a small sample of point probes of the failures may be helpful in boosting the power of connectivity probes as adding a sample of I informs how many nodes to fail and may help to idenfity the true disaster scenario o. Thus, additionally to $\mathcal{Q}_c$ we assume that a small set of (costlier) point queries $\mathcal{Q}_I \subseteq I$ (a sample of actual components that have failed) is also available. For the ease of modeling, we assume that edges $e \in \mathcal{Q}_I$ are sampled uniformly at random with probability γ_I [4]. In the rest of the paper, we refer to $\mathcal{Q}_I$ as the failure probe set and refer to both $\mathcal{Q}_I$ and $\mathcal{Q}_c$ together as the joint probes.

C. MDL

Note that we do not know apriori how many failures are present in the system. Hence, intuitively, to formulate the problem above, we need a model selection framework which finds this automatically in a principled fashion. To this end, we propose to use the information-theoretic Minimum Description Length (MDL) principle [8]. We use two-part MDL, also known as the sender-receiver framework as we are specifically interested in the model. Our goal is to transmit the given set of probes $\mathcal{Q} = \mathcal{Q}_c \cup \mathcal{Q}_I$ (the 'data') from sender to receiver in the least cost (in bits), by assuming that both of them know the layout of the network G, the sampling rates γ_c and γ_I, the possible failure sources $\mathcal{O}$ and the conditional failure parobabilities $F's$. The MDL cost function consists of two parts: (a) Model cost includes the complexity of the selected model that explains the state of the network; and (b) Data cost that represents the cost of sending the given probe data $\mathcal{Q}$ given the model chosen. Formally, given a set of models $\mathcal{M}$, MDL claims the best model M^* is the one that minimizes $\mathcal{L}(M) + \mathcal{L}(\mathcal{D}|M)$, in which $\mathcal{L}(M)$ is the model-cost (length in bits to describe model M), and $\mathcal{L}(\mathcal{D}|M)$ is the data-cost (the length in bits to describe $\mathcal{D}$ using M).

TABLE I
TABLE OF SYMBOLS

Symbol	Definition
V_S, V_T, V_D	Supply, transshipment, and demand node set, respectively
$\mathcal{O}$	Set of disaster scenarios
I	Set of failed edges (failure set)
$F(e\|o)$	Failure probability of edge e given disaster o
$\mathcal{S}$	Set of serviced nodes (serviced set)
γ_c, γ_I	Probability for (uniformly) sampling probes from sets $\mathcal{S}$ and I, respectively
$\mathcal{Q}_c$	Connectivity probe set (connectivity queries) sampled from set $\mathcal{S}$
$\mathcal{Q}_I$	Point probe set (point queries) sampled from set I
$h(G, \mathcal{S}, I)$	Indicates if the concurrence of G, $\mathcal{S}$, and I is possible (returns 1) or not (returns 0)

D. Model Space and Cost

The first step in our formulation is to identify the right model *space*. Our failure process states that the failure set I is induced due to a particular disaster scenario $o \in \mathcal{O}$. Hence, we include both o and I in the model. In real networks different disasters induce failures in totally different regions of the network. Hence, marginalizing over different scenarios and picking the most likely failure set I over all disasters [4] does not work well.

We want to primarily infer only I, but this set by itself is not enough to be the model because there is no direct relationship between I and Q_c; thus, it would not be useful to explain the data. Therefore, we use all three of o, I and $\mathcal{S}$ as our model: $\mathcal{M} = (o, \mathcal{S}, I)$. In other words, we first send the disaster scenario o, then send the true serviced set $\mathcal{S}$, followed by the true failed set I. We then identify the actual joint probes set $\mathcal{Q}_c$ and $\mathcal{Q}_I$ as the data.

The MDL model cost $\mathcal{L}(o, \mathcal{S}, I)$ has the following three components:

$$\mathcal{L}(o, \mathcal{S}, I) = \mathcal{L}(o) + \mathcal{L}(\mathcal{S}|o) + \mathcal{L}(I|\mathcal{S}, o)$$

Using the Shannon-Fano code we have the following.

$$\mathcal{L}(o) = -\log(Pr(o))$$
$$\mathcal{L}(\mathcal{S}|o) = -\log(Pr(\mathcal{S}|o))$$
$$\mathcal{L}(I|\mathcal{S}, o) = -\log(Pr(I|\mathcal{S}, o))$$

Combining and expanding these, we obtain the following.

$$\begin{aligned}&\mathcal{L}(o, \mathcal{S}, I)\\ &= -\log(p(o)) - \log\big(Pr(\mathcal{S}|o)\big) - \log\left(\frac{Pr(\mathcal{S}|I, o)Pr(I|o)}{Pr(\mathcal{S}|o)}\right)\\ &= -\log(p(o)) - \log\big(Pr(\mathcal{S}|I, o)Pr(I|o)\big)\end{aligned}$$

where $Pr(o) = p(o)$. Note from (1) that given I, $\mathcal{S}$ does not depend longer on o. Hence we have, $Pr(\mathcal{S}|I, o) = Pr(\mathcal{S}|I)$. Clearly, $Pr(\mathcal{S}|I)$ is 1 if after removing I from G, the serviced set is exactly $\mathcal{S}$, and 0 otherwise. For $Pr(I|o)$, recall that we assume that edge failures depend only the disaster's effect and are independent from each other, therefore, we can represent $Pr(I|o)$ as a product of failure probabilities of the individual failure. Note that $Pr(\mathcal{S}|I)$ is the probability of set of demand nodes S being connected to the source the given failure set is I. As this probability can only take value of 0 or 1 (indicating whether it is feasible for S to be connected or not), we represent it with a indicator function $h(G, \mathcal{S}, I)$, which we refer to as a feasibility function. Combining all of the above, the final model cost is:

$$\begin{aligned}&\mathcal{L}(o, \mathcal{S}, I) =\\ &-\log\big(p(o)\big) - \log\big(h(G, \mathcal{S}, I)\prod_{e\in I}F(e|o)\prod_{e\in E\setminus I}(1 - F(e|o))\big).\end{aligned} \tag{2}$$

E. Data Cost

Note that MDL needs a data cost as well (otherwise we will choose a model without considering $\mathcal{Q}_I$ and $\mathcal{Q}_c$). In the MDL framework, given the model is already transmitted, we now send the data. For the data cost, we send first the size of $\mathcal{Q}_c$ and $\mathcal{Q}_I$, which are $|\mathcal{Q}_c|$ and $|\mathcal{Q}_I|$, respectively. They followed by the joint queries, i.e. our two probes sets $\mathcal{Q}_c$ and $\mathcal{Q}_I$. Hence, our data cost consists of four terms.

As $\mathcal{Q}_c$ is sampled uniformly from $\mathcal{S}$ with probability γ_c, the size $|\mathcal{Q}_c|$ is send as follows:

$$\begin{aligned}\mathcal{L}(|\mathcal{Q}_c||\mathcal{S}, I, o) &= -\log(Pr(|\mathcal{Q}_c||\mathcal{S}, I, o))\\ &= -\log(\binom{|\mathcal{S}|}{|\mathcal{Q}_c|}\gamma_c^{|\mathcal{Q}_c|}(1-\gamma_c)^{|\mathcal{S}\setminus\mathcal{Q}_c|})\end{aligned}$$

Similarly, for $|\mathcal{Q}_I|$ we have:

$$\begin{aligned}\mathcal{L}(|\mathcal{Q}_I||\mathcal{S}, I, o) &= -\log(Pr(|\mathcal{Q}_I||\mathcal{S}, I, o))\\ &= -\log(\binom{|I|}{|\mathcal{Q}_I|}\gamma_I^{|\mathcal{Q}_I|}(1-\gamma_I)^{|I\setminus\mathcal{Q}_I|})\end{aligned}$$

Then the cost of sending $\mathcal{Q}_c$ and $\mathcal{Q}_I$ in terms of previously sent information is:

$$\begin{aligned}&\mathcal{L}(\mathcal{Q}_c\big|\mathcal{S}, I, |\mathcal{Q}_c|) + \mathcal{L}(\mathcal{Q}_I\big|\mathcal{S}, I, |\mathcal{Q}_I|)\\ &= -\log(Pr(\mathcal{Q}_c\big|\mathcal{S}, I, |\mathcal{Q}_c|)) - \log(Pr(\mathcal{Q}_I\big|\mathcal{S}, I, |\mathcal{Q}_I|))\\ &= -\log(\gamma_c^{|\mathcal{Q}_c|}(1-\gamma_c)^{|\mathcal{S}\setminus\mathcal{Q}_c|}) - \log(\gamma_I^{|\mathcal{Q}_I|}(1-\gamma_I)^{|I\setminus\mathcal{Q}_I|})\end{aligned}$$

Combining the four data cost terms, the total data cost is:

$$\begin{aligned}&= -\log\binom{|I|}{|\mathcal{Q}_I|} - 2|\mathcal{Q}_I|\log(\gamma_I) - 2(|I| - |\mathcal{Q}_I|)\log(1-\gamma_I)\\ &\quad -\log\binom{|\mathcal{S}|}{|\mathcal{Q}_c|} - 2|\mathcal{Q}_c|\log(\gamma_c) - 2(|\mathcal{S}| - |\mathcal{Q}_c|)\log(1-\gamma_c).\end{aligned} \tag{3}$$

Formal Problem Statement. We can now state our problem formally as:

Problem 1: NETPATHSTATE: *Given an undirected graph $G(V, E)$ where $V = V_S \cup V_D \cup V_T$, a many-to-many relation mapping supply nodes V_S to demand nodes V_D, a set of observed serviced nodes $\mathcal{Q}_c$ sampled uniformly at random with probability γ_c from $\mathcal{S}$, and a set of observed failed edges $\mathcal{Q}_I$ sampled uniformly at random with probability γ_I from I, find the complete set of serviced nodes $\mathcal{S}^* \subseteq V_D$ and failed edges $I^* \subseteq E$ (that failed as per the failure model described in Section II), by minimizing the MDL cost function, i.e.*

$$\begin{aligned}&\langle\mathcal{S}^*, I^*\rangle\\ &= \arg\min_{\mathcal{S}, I}\Big\{\\ &-\log\big(p(o)\big) - \log\big(h(G, \mathcal{S}, I)\prod_{e\in I}F(e|o)\prod_{e\in E\setminus I}(1 - F(e|o))\big)\\ &-\log\binom{|I|}{|\mathcal{Q}_I|} - 2|\mathcal{Q}_I|\log(\gamma_I) - 2(|I| - |\mathcal{Q}_I|)\log(1-\gamma_I)\\ &-\log\binom{|\mathcal{S}|}{|\mathcal{Q}_c|} - 2|\mathcal{Q}_c|\log(\gamma_c) - 2(|\mathcal{S}| - |\mathcal{Q}_c|)\log(1-\gamma_c)\Big\},\end{aligned}$$

i.e.,

$$\langle\mathcal{S}^*, I^*\rangle = \arg\min_{\mathcal{S}, I}\Big\{\text{Eq. (2)} + \text{Eq. (3)}\Big\} \tag{4}$$

Algorithm for NETPATHSTATE Problem. Solving Problem 1 is very challenging. First of all, the search space of the solution is exponentially large as it involves searching over all possible sets of serviced and failed nodes, and handle the combinatorial explosion of paths. This makes the problem naturally more challenging than past work in point queries [4]. Moreover, since the objective is a function of two interdependent sets, designing any algorithm with performance guarantee is of a major challenge. Formally, we state the following lemma.

Lemma 1: NETPATHSTATE is NP-hard to approximate within an $\Omega(\sqrt{\log\log|V|})$ factor.

Proof. (Sketch) Our proof is by a reduction from the multicut problem, which is defined in the following manner: we are given an instance $H = (V_H, E_H)$, and a set of source-sink pairs $(s_1, t_1), \ldots, (s_k, t_k)$. The goal is to pick the smallest subset $E' \subseteq E$ of edges, such that all the source-sink pairs are disconnected in $G[E \setminus E']$, which was shown by [11] to be NP-hard to approximate within an $\Omega(\sqrt{\log\log|V_H|})$ factor, under standard complexity theoretic assumptions. We reduce

Algorithm 1 JOINTPATHMAP

Input: Instance $(G, V, \mathcal{Q}_c, F, \gamma_c, \mathcal{Q}_I, \gamma_I)$
Output: Solution $\hat{I}$ as per Eq. (4)
for $o \in \mathcal{O}$ **do**
  $\hat{I}(o) \leftarrow \mathcal{Q}_I$ {initialize solution set for o}
  while $\exists e \in E \setminus \hat{I}(o)$ that decreases the MDL cost **do**
    {To calculate MDL, first compute $\mathcal{S}$ with BFS in $G \setminus \hat{I}(o)$}
    Select $e \in E \setminus \hat{I}(o)$ that decreases the most the MDL cost
    $\hat{I}(o) \leftarrow \hat{I}(o) \cup \{e\}$ {update solution}
    $\alpha(o) \leftarrow$ MDL cost for updated $\hat{I}(o)$
  end while
end for
Return $\hat{I}(o)$ for o that minimizes $\alpha(o)$

this to an instance of NETPATHSTATE in the following manner. We add a new supply node s, and add m' parallel edges (s, s_i) for each i (this can be changed to non-parallel edges, by adding a new node on each such edge). Finally, we add an additional demand node t_{k+1}, and add the path from s to t_{k+1}, which gives us the graph G. We set $V_D = \{t_1, \ldots, t_{k+1}\}$, and $\mathcal{S} = \mathcal{Q}_c = \{t_{k+1}\}$, with $\gamma_c = 1$. Let M be number of edges in G. We set $F(e|o) = 1/2^M$ for all edges e. Observe that any feasible solution must make all the terminals $\{t_1, \ldots, t_k\}$ disconnected from s. Next, the cost of any feasible solution $I \subset E$ is $\leq M|I| + 1$, because of the choice of $F(\cdot)$. Finally, due to the parallel edges (s, s_i), none of those edges will be picked in the solution. This implies, the solution I picked for this instance is also a multicut in H. Since the cost of the solution remains the same, within a fixed multiplicative factor, the lemma follows. ■

Note: It is worth observing that the corresponding point query only formulation of [4] can be approximated within an additive $O(\log n)$ factor. Additive approximations are, in general, easier than multiplicative approximation guarantees. Therefore, Lemma 1 implies Problem 1 is a harder problem than the point query version, from a complexity standpoint.

NETPATHSTATE is quite challenging as it is related to the multicut problem with multiple paths between source-sink pairs. Since a related multicut problem leverages a greedy strategy [12], we adopt this approach. We describe our algorithm JOINTPATHMAP for NETPATHSTATE next. It iterates over the disaster scenarios in $\mathcal{O}$ to find a failure set for each o. To obtain $\hat{I}(o)$, it starts by initializing it to $\mathcal{Q}_I$. Then, at each iteration, an edge $e \in E \setminus \hat{I}(o)$, whose addition decreases the MDL cost the most, is added to the failure set $\hat{I}(o)$. The process is repeated until no such edge can be found. At the end, the $\hat{I}(o)$ that has the lowest MDL score is returned. The implementation needs the calculation of $\mathcal{S}$ after $\hat{I}(o) \cup \{e\}$ is removed from G, which needs to check which demand nodes has at least one path to a supply node, which is accomplished via breadth-first search (BFS). The complete pseudocode is in Algorithm 1.

Lemma 2: Algorithm JOINTPATHMAP runs in $O(|V|^2|E| + |V||E|^2)$ time.

Proof. Calculating total MDL cost for any solution instance $\hat{I}$ requires a BFS (to calculate the $\hat{\mathcal{S}}$ induced by $\hat{I}$) for each demand node, which, takes $O(|V|^2 + |V||E|)$. The inner loop in lines 5-10 takes $O(|E|)$ and the loop in 3-11 takes $O(|\mathcal{O}|)$. In practice, $|\mathcal{O}| \lll |V|$ is a small constant; therefore, our algorithm runs in $O(|V|^2|E| + |V||E|^2)$. ■

Note that the feasibility function contained in the MDL cost will ensure that the infeasible solutions are not selected. This is because the term $h(G, \mathcal{S}, I) = 0$ for infeasible solutions. Hence, the MDL cost for such solutions will be infinite and they will not be picked by our heuristic. Hence, the algorithm always ensures that $\mathcal{Q}_c \subseteq \hat{\mathcal{S}}$.

IV. EXPERIMENTS

In this section, we empirically test our formulations and algorithms in both single and multiple disaster scenarios.

A. Setup

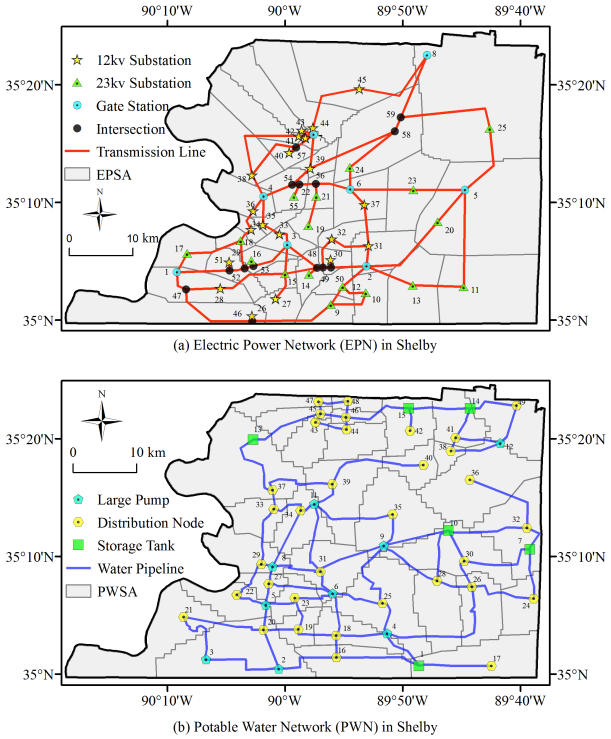
Setup. We implemented the algorithms in Python[1]. We ran our algorithms on a Linux machine with 8 cores with 3.40 GHz and 16GB of RAM. Presented results are the average over 30 different *damage scenarios* simulated as per civil engineering standards (explained in detail in our supplement). Each damage scenario is originated from a single *scenario earthquake* $o \in \mathcal{O}$ (each o is a magnitude and an epicenter) which was randomly selected with probability $p(o)$. JOINTPATHMAP takes roughly 2 seconds to output a solution (for a single damage scenario) for EPN and PWN networks, and roughly 500 seconds for our large network GridKit (see Section IV-B for network specifications). Therefore, our algorithm is fast and scalable.

B. Datasets

We evaluate the performance of our algorithms on the topologies of three real infrastructure networks. For each network, we follow procedures as described in civil engineering literature [9] with structural parameters from [13] to generate multiple geographical correlated failure probabilities for nodes. Edge failure probabilities are calculated from these counterparts.

EPN & PWN. We use two utility networks for Shelby County, TN, USA: electric power network (EPN) and potable water network (PWN), with $|V_{EPN}| = 59$, $|E_{EPN}| = 73$, $|V_{PWN}| = 49$, $|E_{PWN}| = 71$, whose topologies are depicted in Figure 2 and are publicly available in [14]. We adopt the 10 scenario earthquakes that characterize the seismic hazard in the area identified in [9]. This means that once an earthquake happens, its effects on structures can be characterized by one of these 10 scenario earthquakes. Each of the scenario earthquake induce different failure probability-sets (F's) for the network.

[1]Code and data: https://github.com/arodriguezca/JointPathMap



(a) Electric Power Network (EPN) in Shelby

(b) Potable Water Network (PWN) in Shelby

Fig. 2. Skeletonized power and water network topologies in Shelby County, Tennessee.

GridKit. We use the high-voltage power grid for Northern California, obtained from [15], from which we extract the subgraph corresponding to Northern California. It contains $|V_{GK}| = 415$ and $|E_{GK}| = 543$. We generate 50 scenario earthquakes, each of them as follows: epicenter and magnitude are selected randomly, the first from all node locations, and the second from the four values considered in [9].

Table II contains the specific number of demand, source and transshipment nodes for these three networks. As mentioned before, JOINTPATHMAP is fast in these real networks publicly available; therefore, when larger networks become available, our algorithm will apt to run in a suitable time.

TABLE II
DATASETS DETAILS

Network	$\|V\|$	$\|V_S\|$	$\|V_D\|$	$\|V_T\|$	$\|E\|$
EPN	59	9	37	13	73
PWN	49	15	34	0	71
GridKit	415	114	82	219	543

We consider the following factual observation in our experiments. Even though the network topology allows it, in practice, source nodes cannot service *any* demand node, but only the ones close to them that its generative capacity allows. To model this physical constraint, we restrict serviceability of a demand node to the case when there is at least one path of length L to any source. In our experiments, we use $L_{EPN} = 3$, $L_{PWN} = 2$, and $L_{GK} = 4$.

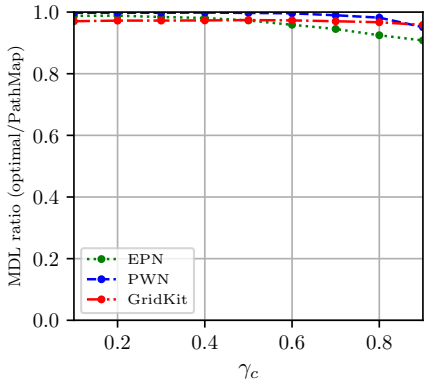


Fig. 3. MDL ratio between optimal and our method JOINTPATHMAP.

C. Baselines and Metrics

We use two baselines, namely MODELCOST and ONLYCONNECTIVITY. MODELCOST is an algorithm that greedily minimizes model cost in Eq. (2), to demonstrate the importance of considering the data cost. ONLYCONNECTIVITY is a baseline that only includes connectivity queries.

To evaluate our algorithms, we use F1-score, for which we need to compute precision and recall. Following the notation in our paper, let I be the true failure set and $\hat{I}$ be our solution set. Our metrics are defined in literature as follows: precision $= |\hat{I} \cap I|/|\hat{I}|$, recall $= |\hat{I} \cap I|/|I|$, and F1-score $=$ precision $\times$ recall$/$(precision $+$ recall). We add the following limit cases to handle empty I or $\hat{I}$: precision is 1 when $\hat{I} = I = \emptyset$ (we correctly identified no failures), and recall is 1 when $|I| = 0$ (there are no failures missed in our solution $\hat{I}$).

D. Results

NETPATHSTATE problem asks to optimize the MDL cost to infer the complete failure set, and JOINTPATHMAP succeeds at this optimization task at low computation cost. This is reflected in our results in Figure 3. We can see that the ratio between optimal MDL and our methods is close to 1. The optimal MDL is calculated via integer linear programming where all paths are listed.

We present the performance of all the methods in Figure 4 in terms of F-1 score. As described before, this score is an average over 30 damage scenarios, each of them coming from a different disaster scenarios o sampled from $\mathcal{O}$ (our scenario earthquakes) with probabilities $p(o)$. As failure probabilities F depend on o, this set of damage scenarios leads to a variety of ground truth failure sets I. As we can observe, JOINTPATHMAP outperforms all the baselines significantly in all three real world networks for all values of γ_c. Poor performance of MODELCOST can be attributed to the fact that it infers the most likely failures while ignoring the observed data. Thus, MODELCOST's result highlights the necessity of the MDL framework.

Similarly, ONLYCONNECTIVITY has low F1-score due to low recall, which indicates that while the inferred failures are mostly correct, they are just a fraction of the total failure set.

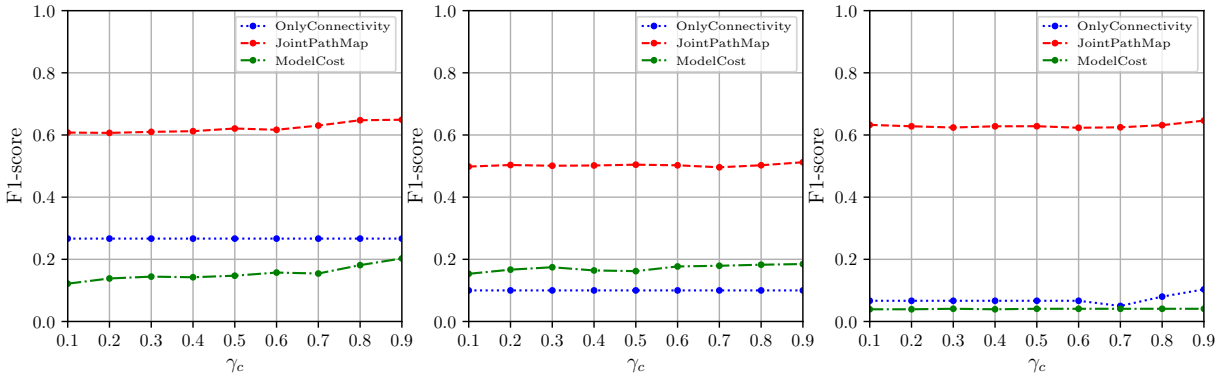


Fig. 4. F1-score for JOINTPATHMAP ($\gamma_I = 0.3$), ONLYCONNECTIVITY and MODELCOST for multiple disaster scenarios. Datasets from left to right: EPN, PWN, and GridKit.

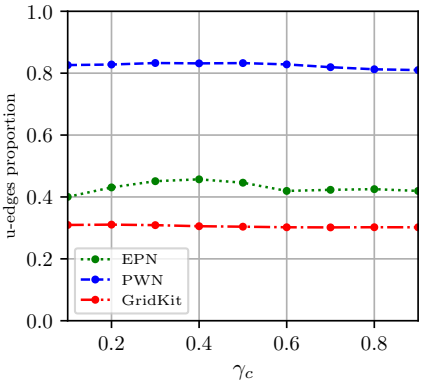


Fig. 5. u-edges proportion is an estimate of the proportion of existing edges in the actual failure set I that are not reflected in the connectivity queries. This proportion is especially high for the PWN, which is at the same time the hardest to solve for ONLYCONNECTIVITY.

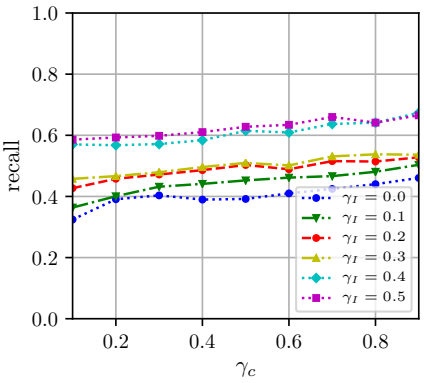


Fig. 6. The effect of increasing γ_I (i.e. increase the sample of failure probes) for EPN dataset. We present recall here because we want to emphasize the ability of our algorithm to discover elements of the failure set I that are not in $\mathcal{Q}_I$.

This is due to the fact that, depending on the redundancy of the network topology, some of the failures are not reflected in the connectivity queries, for which it would be very hard to infer by only using $\mathcal{Q}_c$. To get a sense of how many of 'undiscoverable' edges (u-edges) are present in ground truth failure set I, we add as many edges from I to $\hat{I}$ as long serviceability is not affected (i.e. $\hat{\mathcal{S}}$ does not change). Then, we calculate the proportion of u-edges in I. In Figure 5, we can see that there is a high proportion of u-edges, i.e. edges not reflected in connectivity queries. This indicates there are many failure sets that can produce the same serviced set $\mathcal{S}$; consequently, solely using connectivity probes $\mathcal{Q}_c$ is clearly not sufficient. As shown by the results, JOINTPATHMAP does not suffer from this issue and works well in real datasets. The improved performance is because JOINTPATHMAP incorporates a small set of probes of the failures $\mathcal{Q}_I$, which provides insight about the location of the epicenter thanks to the geographical correlation among failures.

For multiple scenario earthquakes, the algorithm has to infer first the culprit scenario o and then the complete failure set I. In a small network, some of scenario earthquakes are similar and missing the right one may not drastically affect further inference; however, in large networks, scenarios earthquakes are completely different (mainly in epicenter), therefore, without a proper inference of o it is very hard to make a good inference of I. Adding a small sample of failure probes helps to properly identify o and the right expected number of failures, and gives a larger improvement in F1-score. In addition, we can see in Figure 6 the effect of different sample sizes γ_I, where each increment has a different effect, being the most relevant the jump from $\gamma_I = 0.3$ to $\gamma_I = 0.4$.

As additional results, we also present results for a single disaster scenario. In practice, after an earthquake occurs, an estimate of its magnitude and epicenter is released within minutes, but this information is then corrected several times over the next hours. Therefore, for this kind of natural disaster, it is preferrable to handle uncertainty by considering multiple earthquake scenarios as we did in our formulation. However,

once the estimation about magnitude/epicenter is reliable, we can add this extra information to our problem by reducing $\mathcal{O}$ to a single o and making $p(o) = 1$. For our single disaster scenario experiments, we pick the o that has the largest expected number of failures for EPN and PWN; and, for GridKit, as most disaster scenarios lead to similar damage, we select one at random. As expected, when know the disaster scenario o, this extra information helps the inference as we can see in Figure 7. As we consider the scenario which has the most failures for EPN and PWN (the hardest one), the performance of JOINTPATHMAP may be slightly lower than when aggregating the performance over all scenarios.

E. Case Study

Within the single scenario earthquake framework, we study how the failures discovered relate to the ground truth failure set in PWN. In Figure 8, we observe that our algorithm misses four failures, from which three of them are actually undiscoverable (u-edges). In other words, these failures do not impede any demand node (circle) to be serviced by source nodes (squares) as it can be recognized from the figure. Serviceability in this network is restricted to available paths of length 2 within source and demand node. Consequently, in this network, there are five demand nodes non-serviced (marked with a red X), for which our algorithms found the right failures to disconnect for four of them. In fact, the two failed edges found (dashed blue edges) are the reason why three demand nodes are non-serviced. Therefore, repairing them is of utmost importance to recover service in most of the non-serviced demand nodes. This exemplifies how JOINTPATHMAP finds the truly important informative edges to restore serviceability.

V. RELATED WORK

Critical Infrastructure Networks: Data mining critical infrastructure has been gaining much research interest in recent times [16]. These include works in modeling [17], dependency inference [18], and finding hotspots [5]. Chen et al. [18] proposed collective collaborative filtering based approach for inferring dependency between various interconnected critical infrastructure networks. Similarly Chen et al. [5] proposed cascade based failure model in heterogeneous critical infrastructure networks to identify most critical nodes in the system. A more closely related line of work studies failure inference on infrastructure networks. [19] proposed an algorithm to infer congested intersections given the road networks and usage data. Similarly [4] proposed a near-optimal MDL-based approach to infer failures, but given point probes.

Network Tomography: This involves inferring the network states based on end-to-end network measurements. These measurements, available in the Internet and other communication networks, measure delay across signals transmitted by the probes, which is information that can be used to infer delays on links [20] and network topology [3], [21]. Xia and Tse [20] propose an expectation maximization algorithm to infer delays on links from performing a great quantity of end-to-end measurements. Jin et al. [3] proposed a heuristic to infer approximate topology based on end-to-end measurements with the goal of reducing the number of such measurements while maintaining accuracy. Anandkumar et al. [21] study inferring the topology of sparse networks. They evaluate their algorithm with two routing models and find that they only require a sub-linear number of participant nodes selected uniformly at random.

Epidemic Inference: Another closely related field is reverse engineering an epidemic outbreak over a network given partially observed infections. Shah et al proposed a MLE based approach to detect the source of computer virus spreading over a computer network [2]. Similarly, [22], [23] leverage MDL based approach to detect both the source and missing infections in networks. These works assume that the infection/virus propagation over the network takes a stochastic form often modelled using a variant of the popular SIR model. On the other hand, a different set of works [24]–[26] model the interaction among individuals as a temporal or static time expanded network and treat the combination of the observed infections and the time as a node in the network. The inference is typically done by solving temporally constrained Steiner-tree problem where the terminals are the observed infections.

VI. CONCLUSIONS

We investigated the problem of inferring network states given connectivity queries, a piece of information more practical to obtain than a direct sample of network states (point queries). Our empirical observations indicate that redundancies, common in infrastructure networks, cause a significant proportion of states to be undiscoverable by connectivity probes. Therefore, we propose adding to the connectivity queries a small sample of point queries to exploit relationships among network states, which proved to be valuable for boosting the power of connectivity queries. We formulated the novel problem with MDL and developed a scalable greedy algorithm that consistently optimizes MDL cost close to optimal in extensive experiments. These experiments are performed in real networks and principled simulations. Future work can focus on extending to node failures and using noisy probe data. Other interesting future work is extending our formulation to other applications of connectivity probes such as LAN networks and industrial communication networks.

ACKNOWLEDGEMENTS

This paper is based on work partially supported by the NSF (Expeditions CCF-1918770, CAREER IIS-2028586, RAPID IIS-2027862, Medium IIS-1955883, NRT DGE-1545362), CDC MInD program, ORNL, funds/computing resources from Georgia Tech, and the National Institute of Standards and Technology through the Center for Risk-Based Community Resilience Planning [70NANB15H044, 70NANB20H008]. A.V.'s work was partially supported by NSF grants CRISP 2.0 Grant 1832587, CMMI-1745207, IIS-1633028 and DTRA subcontract/ARA S-D00189-15-TO-01-UVA.

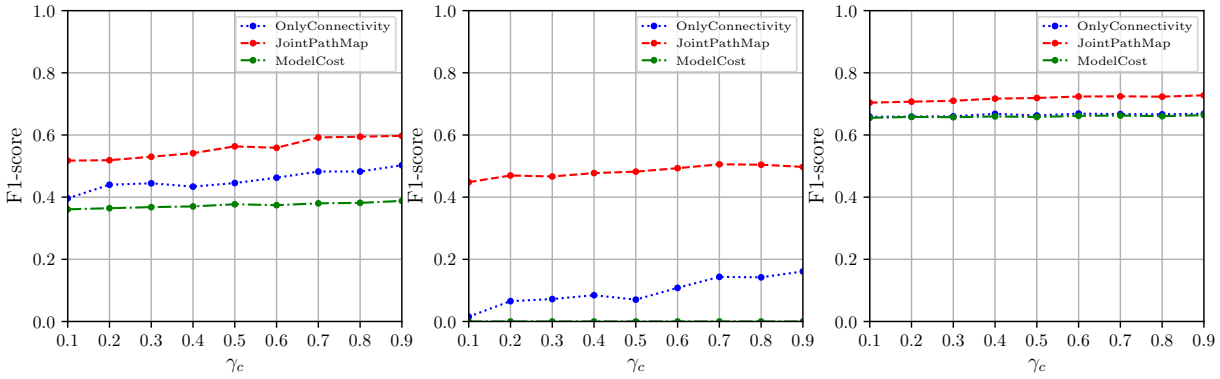


Fig. 7. F1-score for JOINTPATHMAP ($\gamma_I = 0.3$), ONLYCONNECTIVITY and MODELCOST for a single disaster scenario, i.e. o is known. Datasets from left to right: EPN, PWN, and GridKit. MODELCOST returns empty solutions ($\hat{I} = \emptyset$) for PWN because all F's are below 0.5 (i.e. are more likely to remain functional), but the actual expected failure set size is ≈ 10. Therefore, as its recall is 0, its F1-score is constantly 0.

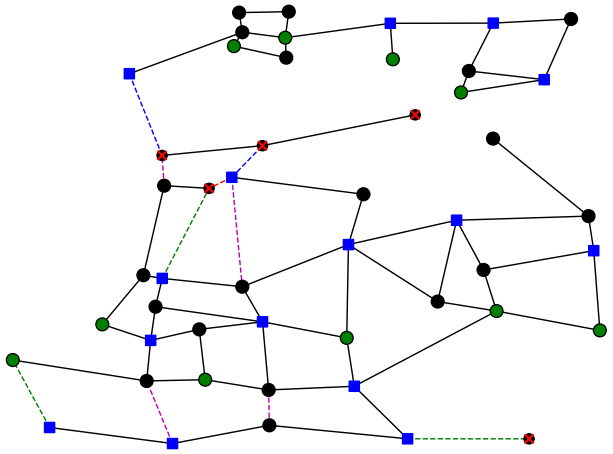


Fig. 8. Visualization of I, $\mathcal{S}$ and $\hat{I}$ in PWN for $\gamma_c = \gamma_I = 0.3$. Source nodes are depicted as squares and demand nodes as circles (there are no transshipment nodes). Probes $\mathcal{Q}_c$ and $\mathcal{Q}_I$ are represented as green nodes and edges respectively. Ground truth non-serviced demand nodes have a red X. Ground truth failed edges are represented as dotted lines. Blue edges are predicted failures by JOINTPATHMAP. Red edges are discoverable failures missed by JOINTPATHMAP. Purple edges are u-edges.

REFERENCES

[1] C. Gomez, A. D. González, H. Baroud, and C. D. Bedoya-Motta, "Integrating Operational and Organizational Aspects in Interdependent Infrastructure Network Recovery," *Risk Analysis*, vol. 39, no. 9, pp. 1913–1929, 2019.

[2] D. Shah and T. Zaman, "Detecting sources of computer viruses in networks: theory and experiment," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 1. ACM, 2010, pp. 203–214.

[3] X. Jin, W. Yiu, G. S. H. Chan, and Y. Wang, "Network topology inference based on end-to-end measurements," *IEEE J-SAC*, vol. 24, no. 12, pp. 2182–2195, 2006.

[4] B. Adhikari, P. Rangudu, B. A. Prakash, and A. Vullikanti, "Near-optimal mapping of network states using probes," in *SDM 2018*. SIAM, 2018, pp. 108–116.

[5] L. Chen, X. Xu, S. Lee, S. Duan, A. G. Tarditi, S. Chinthavali, and B. A. Prakash, "Hotspots: Failure cascades on heterogeneous critical infrastructure networks," in *CIKM 2017*. ACM, 2017, pp. 1599–1607.

[6] S. McClendon and A. C. Robinson, "Leveraging Geospatially-Oriented Social Media Communications in Disaster Response," *IJISCRAM*, vol. 5, no. 1, pp. 22–40, Jan. 2013.

[7] C. Ji, Y. Wei, and H. V. Poor, "Resilience of Energy Infrastructure and Services: Modeling, Data Analytics, and Metrics," *Proc. IEEE*, vol. 105, no. 7, pp. 1354–1366, 2017.

[8] P. Grünwald, "A tutorial introduction to the minimum description length principle," in *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2004.

[9] T. Adachi and B. R. Ellingwood, "Comparative Assessment of Civil Infrastructure Network Performance under Probabilistic and Scenario Earthquakes," *Journal of Infrastructure Systems*, vol. 16, no. 1, pp. 1–10, 2010.

[10] P. K. Agarwal, A. Efrat, S. K. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, "The resilience of WDM networks to probabilistic geographical failures," in *IEEE-Infocom*, 2011.

[11] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar, "On the hardness of approximating multicut and sparsest-cut," *computational complexity*, vol. 15, no. 2, pp. 94–114, Jun 2006.

[12] Y. Kortsarts, G. Kortsarz, and Z. Nutov, "Greedy approximation algorithms for directed multicuts," *Networks: An International Journal*, vol. 45, no. 4, pp. 214–217, 2005.

[13] FEMA, "Multi-hazard Loss Estimation Methodology: Earthquake Model, Hazus-MH 2.1: Technical Manual," FEMA, Washington, D.C., USA, Tech. Rep., 2013.

[14] W. Zhang, P. Lin, N. Wang, C. Nicholson, and X. Xue, "Probabilistic Prediction of Postdisaster Functionality Loss of Community Building Portfolios Considering Utility Disruptions," *Journal of Structural Engineering*, vol. 144, no. 4, 2018.

[15] B. Wiegmans, "Gridkit: European and north-american extracts [data set]," http://doi.org/10.5281/zenodo.47317, 2016.

[16] A. Tabassum, S. Chinthavali, L. Chen, and B. A. Prakash, "Data mining critical infrastructure systems: Models and tools," *IEEE Intell. Info. Bulletin*, pp. 31–38, 2018.

[17] M. Ouyang, "Review on modeling and simulation of interdependent

critical infrastructure systems," *Reliability engineering & System safety*, vol. 121, pp. 43–60, 2014.

[18] C. Chen, H. Tong, L. Xie, L. Ying, and Q. He, "Fascinate: Fast cross-layer dependency inference on multi-layered networks," in *KDD 2016*. ACM, 2016, pp. 765–774.

[19] T. Anwar, C. Liu, H. L. Vu, and M. S. Islam, "Roadrank: Traffic diffusion and influence estimation in dynamic urban road networks," in *CIKM 2015*. ACM, 2015, pp. 1671–1674.

[20] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2235–2248, 2006.

[21] A. Anandkumar, A. Hassidim, and J. Kelner, "Topology discovery of sparse random graphs with few participants," *Random Structures & Algorithms*, vol. 43, no. 1, pp. 16–48, 2013.

[22] B. A. Prakash, J. Vreeken, and C. Faloutsos, "Spotting culprits in epidemics: How many and which ones?" in *ICDM 2012*. IEEE, 2012, pp. 11–20.

[23] S. Sundareisan, J. Vreeken, and B. A. Prakash, "Hidden hazards: Finding missing nodes in large graph epidemics," in *SDM 2015*. SIAM, 2015, pp. 415–423.

[24] P. Rozenshtein, A. Gionis, B. A. Prakash, and J. Vreeken, "Reconstructing an epidemic over time." in *KDD*, 2016, pp. 1835–1844.

[25] H. Xiao, P. Rozenshtein, N. Tatti, and A. Gionis, "Reconstructing a cascade from temporal observations," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 666–674.

[26] H. Xiao, C. Aslay, and A. Gionis, "Robust cascade reconstruction by steiner tree sampling," in *ICDM 2018*. IEEE, 2018, pp. 637–646.

[27] T. Adachi and B. R. Ellingwood, "Serviceability assessment of a municipal water system under spatially correlated seismic intensities," *Computer-Aided Civil and Infrastructure Engineering*, vol. 24, no. 4, pp. 237–248, 2009.

[28] G. R. Toro, N. A. Abrahamson, and J. F. Schneider, "Model of Strong Ground Motions from Earthquakes in Central and Eastern North America: Best Estimates and Uncertainties," *Seismological Research Letters*, vol. 68, no. 1, pp. 41–57, Jan. 1997.

VII. Appendix

A. Data generation

In this paper, we used simulation data developed as per civil engineering standards that encapsulates structural and soil considerations. To generate the different earthquake scenarios for the networks introduced in our paper (EPN and PWN), we followed the procedure described in [9], [27]. First, we used the attenuation equation (Eq. (5)) of [28] to determine the median peak ground acceleration (PGA) on each location, which is associated to the seismic magnitude M_w and epicentral distance R.

$$\begin{aligned}\ln(\text{median PGA}) &= 2.2 + 0.81(M_w - 6.0) \\ &- 1.27\ln(\sqrt{R^2+9.3^2}) - 0.0021\sqrt{R^2+9.3^2} \\ &+ 0.11\ \max\left[\ln\left(\frac{R}{100}\right), 0.0\right] \end{aligned} \quad (5)$$

In particular, we estimated the PGA at the location of each component in the network using simulated seismic events consistent with [9], [27]. Then, in order to determine the failure probabilities associated with each component, we used the fragility curves described in [13].

TABLE III

Median PGA at which the components of the studied networks reach extensive damage, thus, considered failed. Intersection nodes in EPN are considered invulnerable.

Network	Component Type	Median (g)	β_{sd}
EPN	Gate station	0.47	0.40
	23-kV substation	0.70	0.40
	12-kV substation	0.90	0.45
PWN	Storage tanks	0.68	0.75
	Pumping stations	0.80	0.80
	Distribution nodes	1.15	0.60

In [13], it is assumed that given S_d — a spectral displacement or other relevant Potential Earth Science Hazard (PESH) parameter — the probability associated with being in or exceeding a given damage state ds can be modeled using a cumulative lognormal distribution, as described in Eq. (6). In Eq. (6), $\bar{S}_{d,ds}$ is the median value of spectral displacement or PESH parameter at which the building or component of interest reaches the damage state ds, β_{sd} is the standard deviation of the natural logarithm of spectral displacement or PESH parameter for damage state ds, and Φ is the standard normal cumulative distribution function [13],

$$P[ds|S_d] = \Phi\left[\frac{1}{\beta_{sd}}\ln\left(\frac{S_d}{\bar{S}_{d,ds}}\right)\right]. \quad (6)$$

For our particular case study, the PESH used is the PGA [13] at the location of each component of the EPN and PWN, calculated with Eq. (5). The values of β_{sd} and the median PGA at which the components of the studied networks reach a damage state ds consistent with complete or extensive damage (depending on the component type) are described in Table III [9], [13], [27]. We consider a component in this damage state as failed. Note that these components are nodes in our networks.

After calculating the probability of failure of each node, we calculate the edge failure probabilities. We assume the failure probability of an edge depends on its incident nodes (if one of the nodes fails, the edge fails). Therefore, $F(e|o) = F(n_1|o) + F(n_2|o) - F(n_1|o)F(n_2|o)$, where $e = (n_1, n_2)$. Finally, having these failure probabilities, we used Monte-Carlo simulation to generate the 30 earthquake-consistent damage scenarios used in this paper.