Critical Flow Centrality Measures on Interdependent Networks with Time-Varying Demands

James Bryan Williams
Department of Computer Science,
University of Toronto
jamesbw@cs.toronto.edu

October 2019

Abstract

This paper applies centrality measures based on critical flows to interdependent networks in which demands vary over time. These component importance measures combine classical flow centrality with concepts from critical infrastructure protection. In a previous work, the utility of critical flows was demonstrated on a flow network that represented a single infrastructure system with static demands. The present work considers the setting where multiple infrastructure systems are interlinked through physical and geospatial dependencies, and where demands for resources vary over time. Since critical flow measures are specific to flow patterns, and not to static network topology, the identification of critical elements requires statistical means. This paper demonstrates one such approach, leaving avenues open for future research. It also presents two forms of reliability analysis based on critical flows — a composite measure using edge reliability, and component failure/degradation analysis.

1 Introduction

Residents of cities depend on infrastructure systems to deliver not only physical resources such as water and gas, but also a range of social goods ranging from education to healthcare. Disruptions in the delivery of essential services can have extremely deleterious consequences for both individual residents and a city as a whole. As a result, understanding the reliability of urban infrastructure systems should be considered a priority for municipal governments.

Infrastructure systems can be disrupted in numerous ways, including deliberate attacks, component failures, and natural disasters. Much of the existing research on critical infrastructure protection, for instance, has focused on protecting infrastructures against damage due to extreme weather or deliberate attacks [27, 13, 28]. Component failure has been studied extensively in the field of reliability engineering (e.g., [8]) and in the various

engineering disciplines (e.g., water [15], drainage [31], electricity [12], telecommunications [40], and transportation [49]). Disruption of networks has also been considered in operations research (e.g., [50]), computer science (e.g., [54, 52]), network reliability (e.g., [22, 11]), graph theory (e.g., [44, 23]), and network science (e.g., [17]).

This work offers a somewhat different perspective. It is not concerned with disruptive events that are short-lived in duration; instead, it focuses on the behaviour of an infrastructure system as it responds to changes in demand or capacity over time. For instance, the degradation of system components can reduce system capacity over longer time horizons, while population growth and the introduction of high density neighborhoods can result in increasing demand for resources. Stated simply, this work is inspired less by terrorism and natural disasters than it is by the evolution of cities.

The author's previous paper [53] introduced a new set of *component importance measures* ("CIMs") that combine flow centrality and location criticality in the context of supply/demand networks. In this approach, the flow through a component is deemed 'critical' to the extent that it delivers resources to critical locations. For reasons of brevity, the author limited the demonstration to a single infrastructure system with static demands.

The current paper extends that previous work to interdependent networks in which demands vary in time. Since the goal of the overall project is to provide urban planners and municipal engineering staff members with a means of reasoning about the long term impacts of changes to urban infrastructure, the ability to represent changing demand patterns is absolutely necessary. For brevity this paper considers only geospatial and physical dependencies, but the method can be extended to consider other forms of connection between infrastructure systems.

The structure of this paper is as follows. Section 2 provides background on interdependent networks. Section 3 introduces the methodology used in this paper, including the network representation. Section 4 provides an evaluation of the methodology on a small model of a city. Section 5 discusses two forms of reliability analysis that can be performed with critical flow techniques. The paper closes with suggestions for future research.

2 Background

Although infrastructure systems are often studied in isolation in the specific engineering disciplines, they are typically coupled to the extent that the failure of components in one system can cause failures in connected systems [10]. In order to understand potential failure modes — as well as properties like vulnerability and resilience — system engineers require methods that represent urban infrastructure as a set of *interdependent systems*.

One salient reason for modeling interdependencies is that connected infrastructure systems are typically more fragile than solitary systems [51]. Not only do such systems have additional failure modes (e.g., *cascading failures* [56]), but the impact of such failures is difficult to predict. For example, water distribution systems impose much greater cascading damage on other systems than they receive in return [55], and they seem

to display a greater propensity to initiate cascading failure in other systems [24].

Various research communities have advocated an integrated view of infrastructure systems, and a growing body of work is available on interdependencies (e.g., [36, 21]). For instance, homeland security initiatives following the September 11th terrorist attacks in the United States spurred numerous efforts addressing infrastructure interdependencies (e.g., [39]). Overviews of techniques for the modeling and simulating interdependent critical infrastructure systems may be found in several places, including [35].

2.1 Requirements for Interdependent Models

In modeling integrated infrastructure systems, a static view is not sufficient. As argued by Amin [2], management of disruption (and prevention of cascading failures) requires an understanding of system dynamics as well as methods of distributed control. CIMs that are based solely on structural properties (e.g., betweenness centrality [32]) should be used alongside measures that account for system dynamics.

Furthermore, any model used to study the disruption of interdependent infrastructures needs to support two different perspectives [26]: (1) a 'system-of-systems' view that focuses on dependencies, and; (2) a traditional view of each individual system that is familiar to managers/specialists. In the context of this paper, this implies that any useful method for urban infrastructure analysis should offer perspectives suitable for specialists, including urban planners.

One useful means of accomplishing this goal is to deploy such models as modules for interactive *geographic information systems* ("GIS") software. Commercial GIS packages such as ESRI ArcGIS are widely used by municipal governments. GIS software is also widely understood by practitioners from a variety of professional disciplines. Lastly, the critical information protection community has begun to use GIS as a platform for resilience and vulnerability analysis [7].

No matter what deployment method is chosen, the task of translating abstract modeling methodologies into working artifacts is made difficult by at least two major constraints. First, infrastructure systems in many countries (e.g., the United States power grid) are not under the control of a single entity; rather, numerous organizations exist with rights (e.g., ownership) over portions of the system [2]. Second, data on infrastructure systems is not readily available, and particularly not in a form that permits detailed analysis of interdependencies. The fact that municipalities lack detailed information on infrastructure assets has motivated some researchers to develop techniques for inferring asset locations from proxy data sources (e.g., [29, 30]).

2.2 Interdependent Networks

Of particular relevance to this work is the literature (scattered across numerous research communities) on *interdependent networks* (a.k.a., 'interconnected', 'multi-plex' or 'multi-level' [21]). This area has received increasing amounts of attention in the last 5 years,

particularly from the physics and network science communities. A recent survey paper can be found in [9], while books on the topic are readily available (e.g., [38, 16, 21, 3, 4]).

To fix terminology, define network A to be **dependent** on network B if the state of B can influence the state of A [41] (see also [33]). Dependencies can take many forms [24]: (1) **physical**, in which the state of A is affected by the material outputs/flows of B; (2) **geospatial**, in which certain components of A and B are in such close spatial proximity such that local events can affect both networks; (3) **informational**, in which A and B are connected by *information and communications technology* ("ICT"); (4) **social**, in which A affects B along social dimensions; (5) **procedural**, where A affects B on the basis of organizational or regulatory structures, and; (6) **financial**, where market conditions, financial crises and other economic events allow one network to affect another. (Alternative classifications appear in [26, 37]).

2.3 Finding Critical Components in Interdependent Networks

Numerous researchers have proposed methods for identifying critical components in interdependent networks. Typical examples are described below:

- Apostolakis and Lemon [5] evaluate the vulnerability of geospatially interdependent infrastructure systems (gas, water, electric) by identifying **critical locations** — geographical points that are susceptible to attack. Each system is represented as a directed network in which vertices can represent not just junctions but also physical features (e.g., manhole covers). Co-location of assets (e.g., shared service tunnels) is modeled by allowing vertices from one graph to appear in another. (Physical dependencies, such as the use of electricity by the water system, are not modeled).

  In their approach, a set of attack scenarios is identified and the networks are analyzed in order to identify *minimal cut sets* (see [12]). The resulting vulnerabilities are prioritized by: (1) the degree to which the targets are accessible to the attacker (i.e., susceptibility), and; (2) the value of the targets from the standpoint of the decision-maker, calculated by summing their expected disutilities. The susceptibility and value are combined to yield a **vulnerability category** — one of five colors ranging from green to red.

- Lee et al. [26] provide a method for prioritizing service restoration activities in an interdependent system-of-systems. Each independent system is represented as a flow network that carries commodities, composed of edges and vertices that may both have capacity constraints. Dependencies are modeled as additional constraints in a mixed integer network flow model. In addition to geospatial and physical dependencies, they allow *shared dependencies* (i.e., for multi-commodity flow networks) and *exclusive-or dependencies* (i.e., to allow flow on a multi-commodity network to be restricted to one type of commodity at a time).

- Duenas-Osorio et al. [18] study the interdependency of electricity and water systems from a topological standpoint. Both geospatial and physical dependencies are modeled, with the water system requiring electricity for pumps, lift stations, and control units. Conditional probability distributions are used to model potential failures of water system components given failure of electricity system components. Three types of vertex removal strategies are used to model disruptions; for each such disruption, a set of metrics are calculated: (1) nodal degree; (2) characteristic path length [25]; (3) clustering coefficient [32], and; redundancy ratio. Flows of water or electricity are not modeled.

- Buldyrev et al. [10] examine the impact of electricity system disruptions on the internet. Geospatial dependencies are modeled by assigning each internet server to the closest power station. Disruptions are initiated by removing power stations and tracking resulting nodal failures — in particular, a node v is considered to be failed if: (1) all of v's neighboring nodes are failed, or; (2) the geospatially coupled node in the electricity network is failed. Nodes are ranked according to the consequences of removal. The authors argue that disruption of a small number of nodes in the electricity system is sufficient to provide cascading failures in the internet network.

- Galvan and Agarwal [20] perform vulnerability analysis on interdependent infrastructures by examining the impact of disruptions. Each infrastructure is represented as a flow network with a unique resource type. In each iteration of the analysis, a single node is selected for failure (disruption). After recomputing the flow solution, the algorithm identifies every node that is in violation of capacity constraints. These latter nodes are then disabled and the process repeats itself until no more failures occur.

  The authors introduce a new vulnerability metric X_1, defined as the fraction of nodes that fail after the first step of the cascading failure process. After using X_1 to rank nodes, they compare the results against traditional centrality measures (i.e., nodal degree, the flow value for the non-disrupted solution, and network efficiency).

- Svendsen and Wolthusen examine interdependent critical infrastructures in a series of papers [46, 45, 47, 48]. Their models represent multiple concurrent types of dependencies, categorized at a high level into *storable* and *non-storable* types. Each vertex v in a network can act as a producer or consumer of up to m different resources, and for each such resource v has a corresponding buffer. The authors investigate numerous issues, including the behaviour of systems with cyclic interdependencies.

3 Methodology

The work described in this paper can be viewed as a further development of prior art that incorporates insights from critical infrastructure protection (e.g., [5]) and urban planning. It provides a method to identify critical components in physically and geospatially interdependent [26] flow networks as demands vary over time. Unlike previous work, the method is intended for use in interactive software applications (e.g., GIS software [7]).

3.1 Network Representation

As in the preceding work [53], an infrastructure system is represented as a weighted, capacitated, flow network $G = \langle V, E \rangle$ where G is a set of nodes, $E \subseteq V \times V$ is a set of edges, and:

- each node $v \in G.V$ has Euclidean **coordinate** $\vec{w}(v) = (v_x, v_y, v_z) \in \mathbb{R}^3$, as well as an (optional) capacity constraint $c(v) \in \mathbb{N}$.
- each edge $e = (v_i, v_j) \in G.E$ has a **capacity** $c(e) \in \mathbb{N}$, a **flow** $f(e) \in \mathbb{N}$, and a **length** $l(e) \in \mathbb{R}$ defined as $\|\vec{w}(v_i) - \vec{w}(v_j)\|_2$.

Note that each network G is a *multi-graph* in which multiple edges may connect a given pair of nodes, allowing for redundant (fallback) connections. Bi-directional relationships, cycles, and self-loops are all permitted.

A network G contains both source (supply) and sink (demand) nodes. The set of **source nodes** is $V_S = \{s_1, s_2, \ldots, s_p\} \subseteq V$, and the set of **demand nodes** is $V_D = \{d_1, d_2, \ldots d_k\} \in V$. All other nodes are called *transmission nodes*. A **flow** on G is a real-valued function $f : E \rightarrow \mathbb{R}$ on G's edges that obeys three flow properties:

1. **Capacity Constraints**: for all $e = (v_i, v_j) \in E$, we have $f(e) \leq c(e)$.
2. **Skew Symmetry**: for all $e = (v_i, v_j) \in E$, we have $f((v_i, v_j)) = -f((v_j, v_i))$.
3. **Flow Conservation**: for all transmission nodes $v_t \in V - (V_D \cup V_S)$, we have $\sum_{v \in V} f((v_t, v)) = 0$.

Note that each network carries a single type of resource/commodity, unlike the multi-commodity approach in [47]. The **value of a flow** is defined as the flow exiting the source nodes: $|f| = \sum_{v \in V} \sum_{s \in S} f(s, v)$. While a network with multiple source and sink nodes may be reduced to a network with a single sink and source (see [14]), the explicit representation is used throughout this paper. Note also that any node may supply or demand resources, but for simplicity the example in this paper uses disjoint sets for supply, transmission and demand nodes. (Multi-functional nodes are handled using a standard maximum flow reduction in the network conversion process, described in Section 3.5.1).

3.2 Supply Constraints and Demand Distributions

In contrast to the previous work [53], the model in this paper is *explicitly temporal*. Supply constraints and demand distributions are no longer given as integers, but as discrete *time series* (see [43]). For simplicity, each time series is assumed to be regularly sampled at times $t_i \in T = \mathbb{N}^+ = [0, \infty]$. The model takes two types of time series, represented here as the output of functions:

- Each supply node $v \in V_S$ may be assigned an optional **supply constraint function** $f_v^s(t) : T \rightarrow \mathbb{N}^+$ that gives the maximum amount of resource that may be supplied from v at time t.
- Each demand node $d \in V_D$ has a mandatory **demand function** $\delta_d(t) : T \rightarrow \mathbb{N}^+$ that gives the amount of flow required by node d at time t.

An **assignment** to a network involves specifying demand functions for all demand nodes, and (if desired) supply constraint functions for all supply nodes. From a data perspective, they can be viewed as discrete, regularly sampled time series that may come from either probability models or external data sources (e.g., real world data obtained from smart meters).

In the simple embodiment presented in this paper, computations on the network (e.g., network flow solutions, criticality measures) are performed for each time $t_i \in T$. Values from previous time steps t_k may be used as input for computing values in the current time step t_i (where $t_k < t_i$). This permits the method to represent *delays* in resource utilization.

3.3 Criticality Metrics

As in the previous work [53], a **criticality function** $cr : V_D \rightarrow \mathbb{R}$ maps demand nodes $d \in V_D$ to a criticality rating $cr(d)$. Although three types of criticality function exist (i.e., binary, categorical, and continuous), this paper focuses on the continuous representation in which $cr(d) \in [0.1]$. The **critical flow centrality** ("CFC") of a node $v \in V$ under assignment A is:

$$C^{CF}(v) = \sum_{d \in V_D} c_r(d) E[f_A(v, d)]$$

where $f_A(v_i, d_j)$ is the flow that reaches $d_j \in V_D$ from node $v_i \in V$ under assignment A and $E[f_A(v, d)]$ is its expectation. (A similar definition holds for edges). The CFC is the sum of all expected amounts of flow being delivered via v to the demand nodes, weighted by their criticality. An algorithm for computing the CFC was provided in [53].

3.4 Interdependencies

A *system-of-systems* ("SoS") model consists of a set of k infrastructure systems $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$. As shown in Figure 1, two types of dependencies are permitted between pairs of elements from $\mathcal{S}$:

1. **geospatial dependencies**, which arise when elements from network A are *co-located* with those from network B. Co-location of nodes, for example, is an equivalence relation on $\{v : v \in \bigcup_{S_i \in \mathcal{S}} S_i.V\}$.

   This type of dependency is implemented by storing information on co-located network elements in a secondary table data structure. In an interactive application, these dependencies can be suggested automatically by using location information $\vec{w}(v)$ for a node v.

2. **physical dependencies**, wherein elements in network A require resources flowing through network B. Physical dependencies between nodes, for example, are irreflexive, asymmetric and transitive relations.

Dependencies are represented in higher-level data structures and not directly in the networks themselves. For example, vertices from one network S_i do not appear directly in another network S_j (in contrast to [5]). This design choice makes it easier to integrate disparate modeling methods for each individual infrastructure system (see [42]).

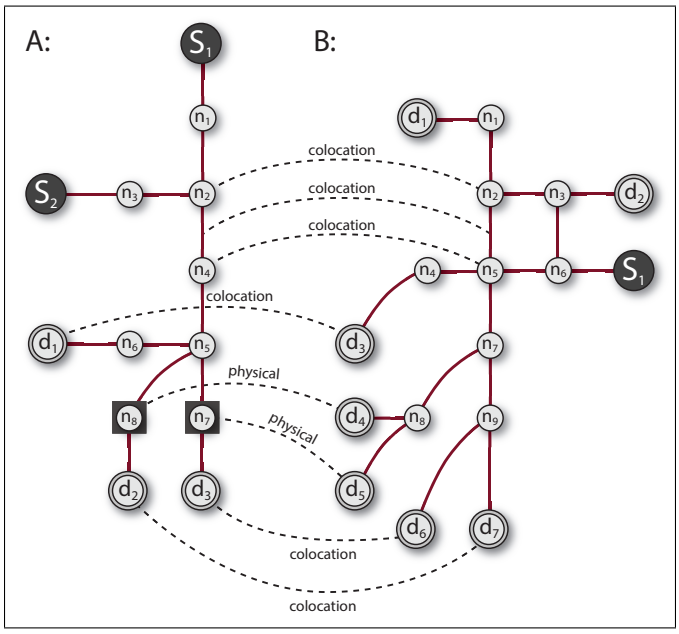


Fig. 1. Two independent infrastructure systems (A) and (B), linked by geospatial and physical (e.g., flow-related) dependencies.

Physical dependencies are represented by *interconnections* between network elements. Referring to **Figure 2**, let S_1 represent a water distribution system, and let S_2 represent an electricity system. A dependency between water node $v_1 \in S_1$ and electricity node $v_2 \in S_2$ is represented by an **interconnection record** $IR(v_1, v_2)$ stored in a secondary data structure. The amount of resource R demanded of S_2 by v_1 (e.g., the amount of electricity required to operate a given water pump) is given by a function $f^R : S_1.V \rightarrow \mathbb{R}$. For instance, a pump at v_1 might demand a constant amount of electricity per unit time, or it may require power proportional to the flow $f(v_1)$ through v_1 (e.g., $f^R(v) = cf(v)$). *Delays* can be accommodated by deferring this demand to later time steps.

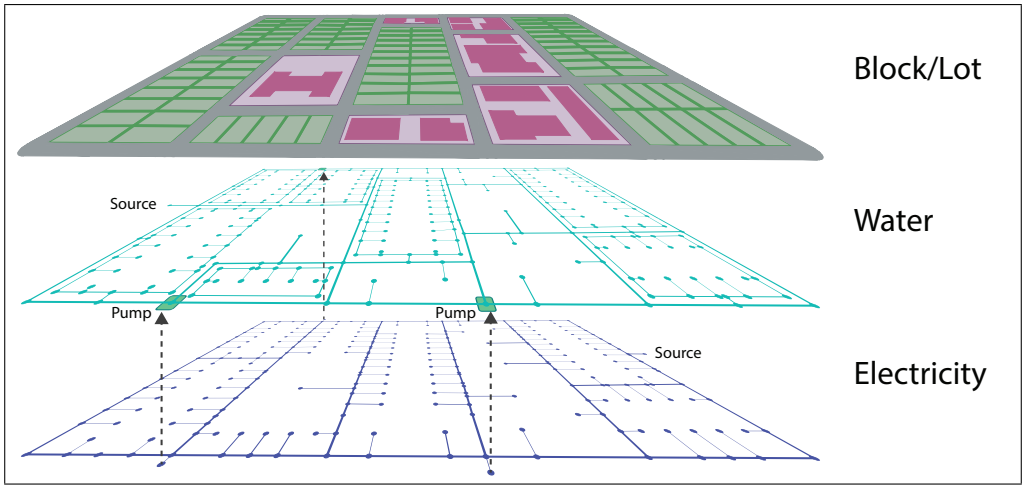


Fig. 2. Infrastructure model with two layers, showing resource flow between water pumps and electricity nodes.

Dependencies between network elements imply dependencies between systems. If an interconnection record exists that maps elements of S_1 to elements of S_2, we say that S_1 is **physically dependent** on S_2, represented as $S_1 \rightarrow S_2$. Mutual dependency between systems makes the computational task more difficult. The methods of Svendsen and Wolthusen (e.g., [47]) accommodate mutual dependencies using multi-commodity flows; however, this approach does not allow for infrastructure-specific network representations and solution methods.

In this paper, the set of physical (resource) dependencies between systems in $\mathcal{S}$ is taken to form a *directed, acyclic graph* ("DAG") $\mathcal{G}$ that can be ordered with a topological sort (see [14]). In contrast, geospatial dependencies are not restricted in such a fashion.

Finally, *criticality ratings* are preserved via physical dependencies. For demand nodes d in S_1 that also have physical dependencies on resources in S_2, the criticality of d in S_1 is associated with the corresponding demands in S_2. For a non-demand node v in S_1, the CFC value for v is associated with the corresponding demands in S_2. This is explained in more detail below.

3.5 An Algorithm for Interdependent Critical Flow Centrality

Critical flow computation in an interdependent network is carried out in an iterative manner. Recalling prior work in [53], the framework for computing the CFC for an individual (single) infrastructure system consists of four elements:

1. a representation of an infrastructure system as a flow network;
2. criticality metadata for nodes (i.e., demand values, criticality ratings);
3. a means of calculating flows;
4. a means of determining the probability that a given network component (node, link) carries flow to a given demand node;

The method presented in the present work proceeds by computing the CFC for each individual infrastructure system in topological order. Dependencies are propagated from one system to the next in each iteration, passing demands from higher-level layers to lower-level ones. **Algorithm 1** provides a high level overview:

Function *ComputeInterdependentCFC*($\mathcal{G}$)
  Data: G, a graph with nodes $V_{\mathcal{G}} = S = \{S_1, S_2, \ldots, S_k\}$ representing individual infrastructure systems, and edges $E_{\mathcal{G}}$ formed from physical dependencies between elements of $V_{\mathcal{G}}$.
  ConvertNetworkRepresentation($\mathcal{G}$)
  Var list $\leftarrow$ *TopologicalSort*($V_{\mathcal{G}}$)
  Var t $\leftarrow$ 0
  foreach $S_i \in$ *list* **do**
    ComputeSingleSystemCFC(S_i)
  end

Algorithm 1: Computing CFC for a set of interdependent infrastructure systems.

3.5.1 Converting Network Representations

To use the techniques from [53] on an SoS model, conversion of network representations is performed to transform each individual network S_i into a format compatible with maximum flow algorithms.

1. Nodes with demands are connected to a *supersink* node (see [14, 1]).
2. Source nodes are connected to a *supersource* node.
3. Nodes in network S_1 that require resources from network S_2 are represented in S_2 by corresponding demand nodes.

In the case of (3), the criticality for the nodes in S_1 is only available after the CFC for all non-demand nodes has been computed. Thus, the full computation for S_1 must be performed before any computations can be performed for S_2. **Figure 3** provides an illustration of network conversion.

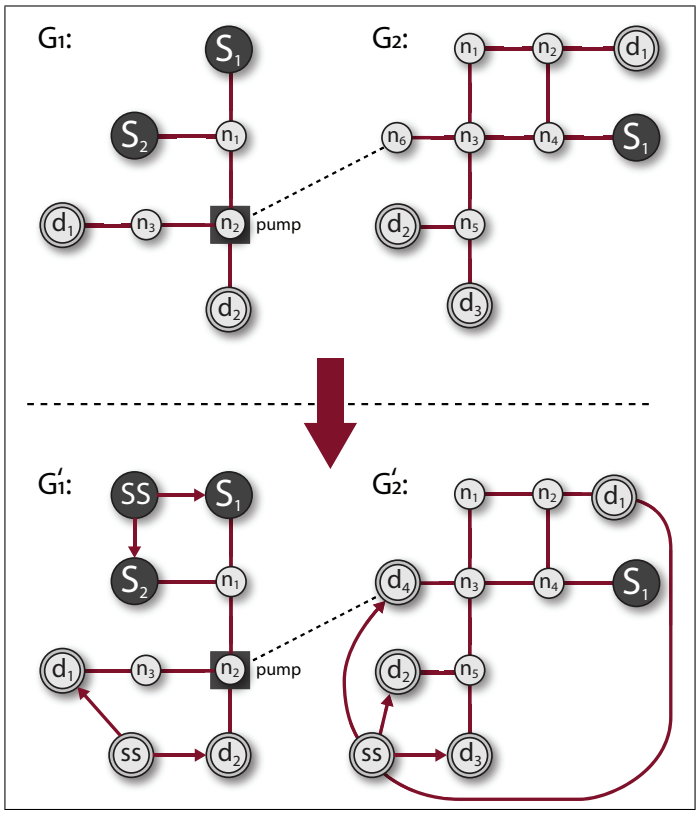


Fig. 3. Two independent infrastructure systems S_1 and S_2, transformed into flow networks suitable for the Edmonds-Karp algorithm. Supersource and supersink nodes ('ss') are added in the usual manner.

3.5.2 Single Layer Computation

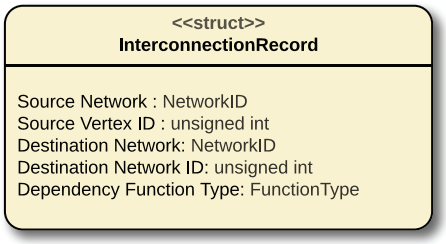
Computation of the CFC for each individual infrastructure system S_i proceeds in two stages: (1) flow values and criticality values are propagated from other layers S_h ($h < i$), and; (2) the CFC for S_i is computed according to the methods in [53]. If layer S_i supplies layer S_h with resources (e.g., it is an electricity network that supplies power to water pumps), then resource demands for S_h appear in S_i's network as sinks with appropriate demands. (The use of topological ordering ensures that S_h's criticality and flow values have been computed before S_i's.)

Algorithm 2 provides an overview of single layer CFC computation. The maximum flow algorithm *ComputeMaxFlow()* and critical flow centrality algorithm *ComputeCFC()* are both exactly as described in [53]. The sole difference from that previous work is the requirement to load propagated demand and criticality values into the model before running those algorithms.

Function *ComputeSingleSystemCFC*(S_i)
    PropagateValues(S_i)
    ComputeMaxFlow(S_i)
    ComputeCFC(S_i)

Algorithm 2: Computing CFC for a set of interdependent infrastructure systems.

Propagation of criticality and flow values proceeds by examining the set of relevant interconnection records $IR(v_1, v_2)$. These records are created by the modeler using the interactive application to connect vertices in different layers of the interdependent system. The implementation of the records includes the following details:



If there exists an interconnection record $IR(v_1 \in S_h, v_2 \in S_i)$, then there is a physical (resource) dependency between systems S_h and S_i. Demand and criticality values for v_1 must be propagated to v_2 before the maximum flow and CFC can be computed for S_i. **Algorithm 3** gives an overview of this process. Criticality values are copied directly, but the amount of resource that must be provided by v_2 to v_1 is determined by a function. In this paper, the resulting demand at v_2 is half of the flow at v_1.

Function *PropagateValues*(S_i)
    foreach *interconnection record* $IR(v_1, v_2)$ **do**
        if $v_2 \in S_i.V$ **then**
            v_2.demand $\leftarrow$ *CalculateResultingDemand*((v_1, v_2))
            v_2.criticality $\leftarrow$ v_1.criticality
        end
    end

Algorithm 3: Propagation of resource demands.

Figure 4 shows water system and electricity system that are interlinked in two locations: pumps near the source of the water system are fed by electricity nodes labelled A and B.

A flow solution was first computed for the water system, yielding flows of 6063 litres and 5973 liters at the pumps. The induced demand at nodes *A* and *B* of the electricity system are 3031 and 2986 units. As in the previous work [53], edges and vertices with no flow are shown in black. The existence of such elements is an artifact of the Edmonds-Karp algorithm [1, 14] used in this simple instantiation, and one that would be corrected by using domain-specific methods (e.g., hydraulic simulation [34]).

Figure 5 shows the CFC values for the same interdependent infrastructure system under the same flow solution. Criticality levels (ranging from 0 to 1) are shown in white font for the buildings. (Lot criticality is fixed at 0.02, and elided for brevity). The electricity nodes *A* and *B* have inherited criticality values of 0.32 and 0.61 from the corresponding pump vertices in the water system; they require flow of 3031 and 2986 units, which the reader can verify by inspection are half of the flow values at the water pump.

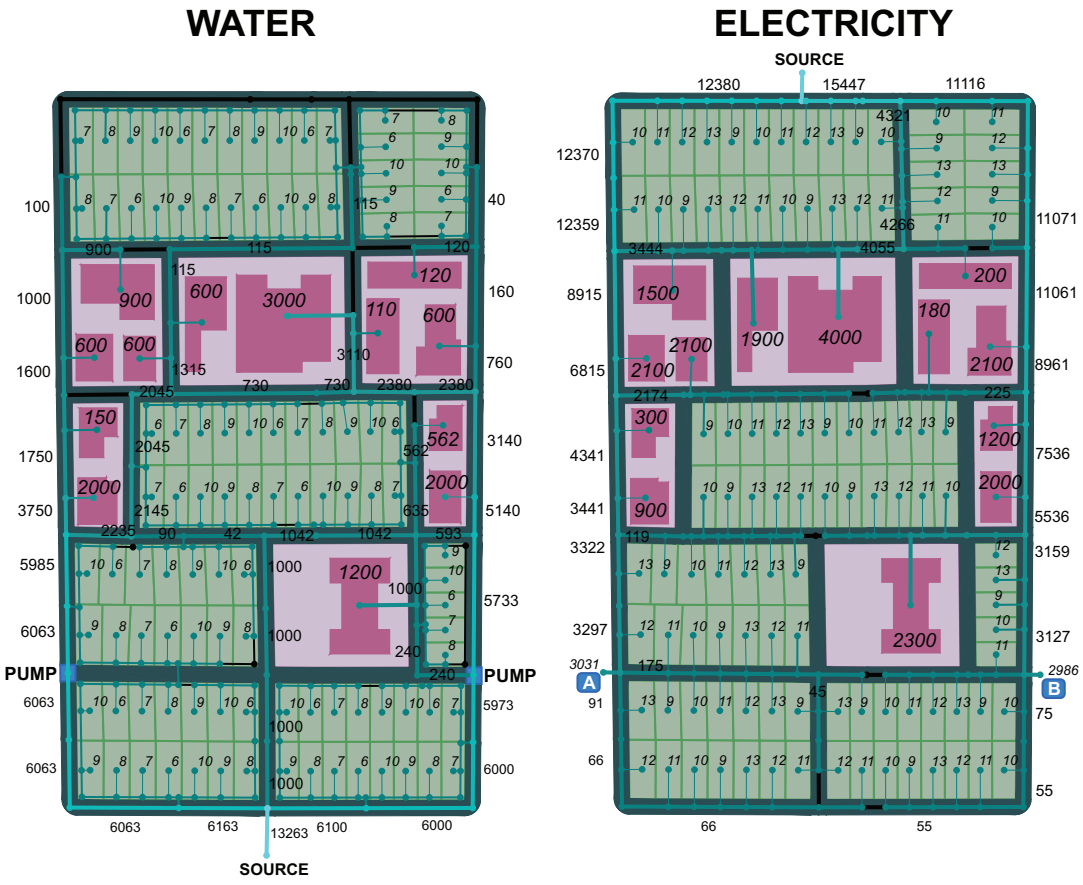


Fig. 4. Flow through an interdependent network. Demand values are in italics while flow values are in regular font. Pumps in the water network are supplied with electricity by nodes A and B. Pumps require electricity proportional to half of their water flow. Black edges/vertices have zero flow.

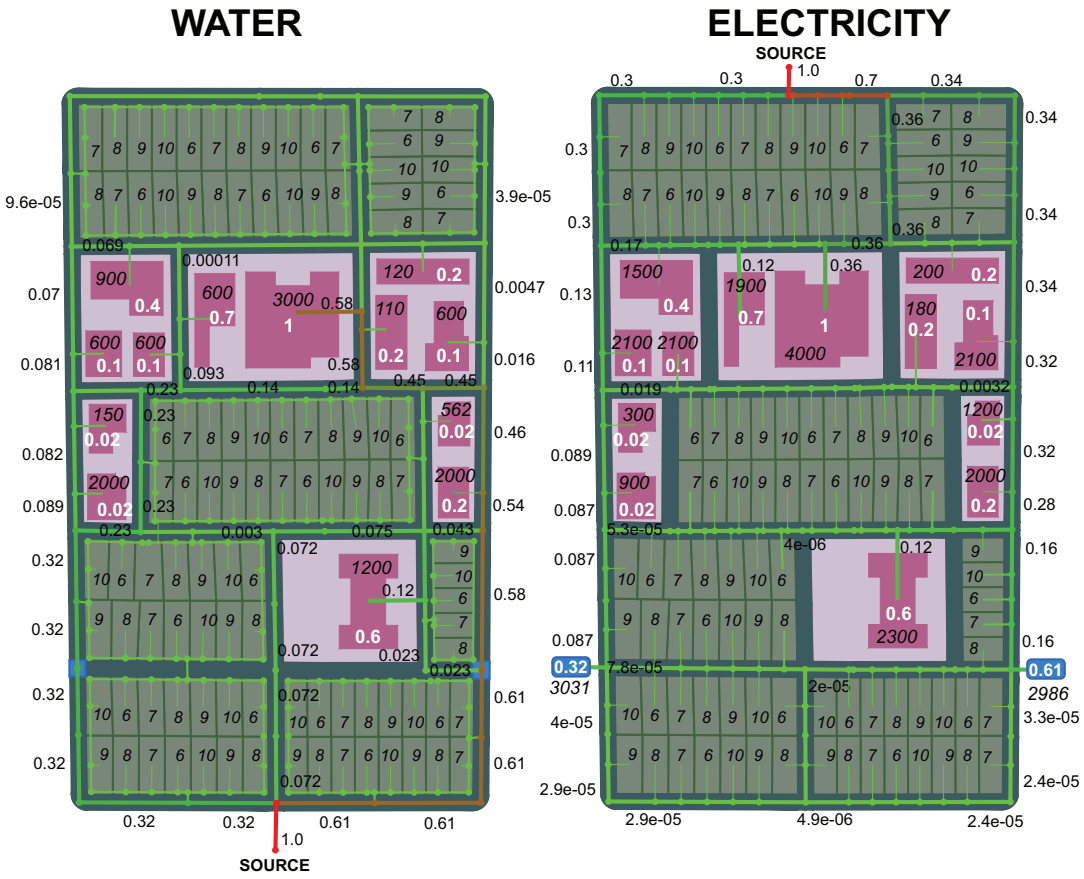


Fig. 5. (Normalized) critical flow centrality, computed from the flows in Figure 4. Demand values are in italics, CFC values are in regular font, and criticality ratings for buildings are in white. The electricity nodes that supply the water pumps are given criticality ratings of 0.32 and 0.61 and demands of 3031 and 2986 via Algorithm 3.

While most of the critical demand in the model is for the hospital (criticality=1.0) and secondary school (criticality=0.6), the pumps create significant critical demand in otherwise non-critical regions of the model. **Figure 5** show that the electricity nodes supplying the pumps carry 16.7% and 8.7% of the total critical flow in the electricity network. Needless to say, it would be a poor decision to co-locate electrical assets with water assets when both are carrying highly critical flow.

Thanks to the propagation of both flow and criticality values from one network to the next, the criticality of the water pumps is appropriately represented in the criticality ratings of the electricity system. While the model used in this paper consists of only two independent systems, it is possible to use the method on interdependent infrastructure models of arbitrary complexity, provided that the interdependencies create a directed, acyclic graph.

3.6 CFC Computation with Time-Varying Demands

The method outlined in [53] provided a means of computing critical flow centrality measures on a static network with integer-valued demands. The CFC values for a network are computed from a given network flow, which in turn is generated from a set of demand values and capacity constraints. CFC measures are therefore based on both network topology and *assignments* (see Section 3.2), as opposed to classical centrality measures (e.g., flow centrality [19]) which are based solely on topology.

Of course, the properties of real-world networks are subject to various forms of dynamics, including demands that vary over time (e.g., due to daily usage patterns) and changing capacity constraints (e.g., due to disruption, failures, or component decay). Additionally, systemic measurement errors or data uncertainty may force modelers to consider a range of demand values, as opposed to simple integers.

One means accounting for these situations is to think of an assignment of demand, capacity and criticality values as a *sample* — each assignment shows one possible state of the network, and the resulting CFC values are measures of criticality for that given state. Evaluation of a system's criticality is performed via a sampling procedure in which multiple assignments are generated, CFC values are computed, and the results are aggregated to form an estimate of component criticality across a wide range of system states.

As an alternative to a sampling approach, one could replace the integer-valued demands with *time series*. This is the approach pursued in this paper, since the goal of this work is to support urban planners and municipal engineers in considering the impact of changes in demand patterns over medium/long time-frames. Each demand node corresponding to a lot/building receives a time series for each resource; in the current paper, time series are assumed to give average hourly demands over a 24-hour day. Demand data for different types of buildings were obtained from several sources (e.g., [6]). Examples of water demands appear in **Figure 6** below:

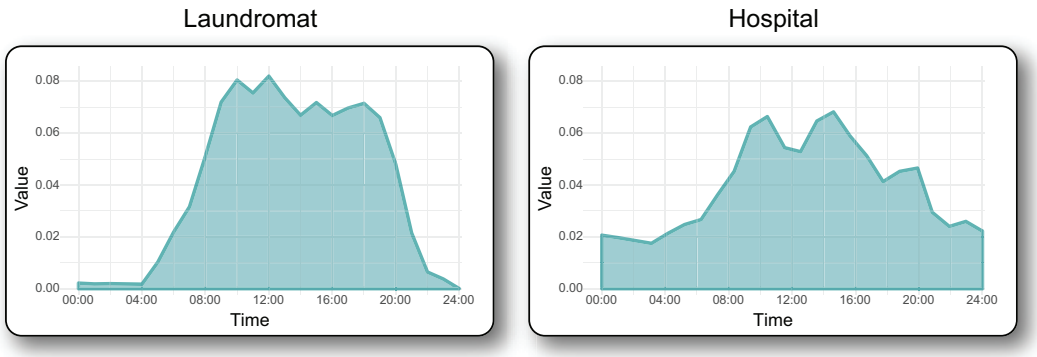


Fig. 6. Hourly time series showing water demands from a laundromat and hospital over an average day. The time series have been normalized to create a probability distribution. For use in the CFC method, these distributions are scaled by average water usage per day.

More precisely, each building/lot in the model is assigned: (1) a time series representing *hourly demand for water*; (2) a time series representing *hourly demand for electricity*, and; (3) a *criticality rating* in the interval $[0, 1]$. Time series data is assigned to buildings according to type (e.g., secondary school, restaurant), while lots are assigned time series drawn from a library of typical residential demand curves. For simplicity, criticality ratings and vertex/edge capacities are assumed to be static, although they could easily be represented with their own time series.

CFC values are computed for each time step $t \in [1, T]$ by loading the relevant time series data for t and executing **Algorithm 1**. An overview of the process is provided in **Algorithm 4**:

Function *ComputeCriticaltyForTimeSeries*($\mathcal{G}$)
  foreach $t \in [1, T]$ **do**
    LoadDemands($\mathcal{G}$, t)
    ComputeInterdependentCFC($\mathcal{G}$)
  end

Algorithm 4: Computing CFC on a system-of-systems with time-varying demands.

Upon termination of this procedure, each node and edge in the interdependent system has a set of CFC values — one for each time step — that can be used in statistical analysis.

4 Evaluation

Figure 7 shows a graph of *critical flow centrality* ("CFC") [53] values for the water network's edges over the full 24-hour cycle:

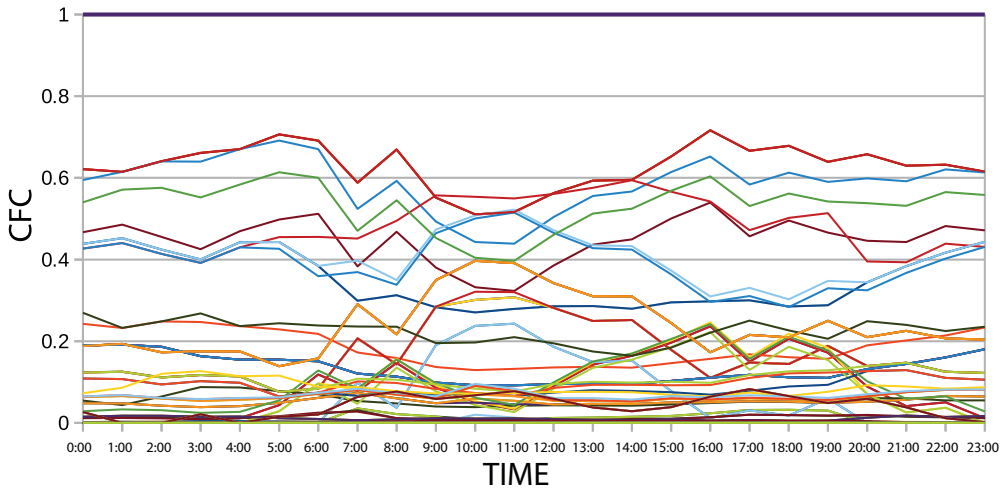


Fig. 7. CFC values for each edge in the water network.

The edge with a constant criticality rating of 1 is the lone edge incident to the source/reservoir. In general, the edges with significant criticality values tend to remain critical throughout the 24-hour cycle, with interesting behaviour happening during the middle of the day. Low criticality nodes become more critical during mid-day, when significant water demand begins to push capacity constraints.

In contrast, the edges of the electricity network display a more stable distribution. In **Figure 8**, one can clearly see that there are fewer intersections between lines in the plot of electricity edge criticality values. The edge to the single source node again has a constant criticality rating of 1, and the fluctuation in criticality values of other major edges is much less pronounced. This is likely a consequence of the fact that the demand on the electricity network does not tend to push capacity constraints as much as the demand on the water network.

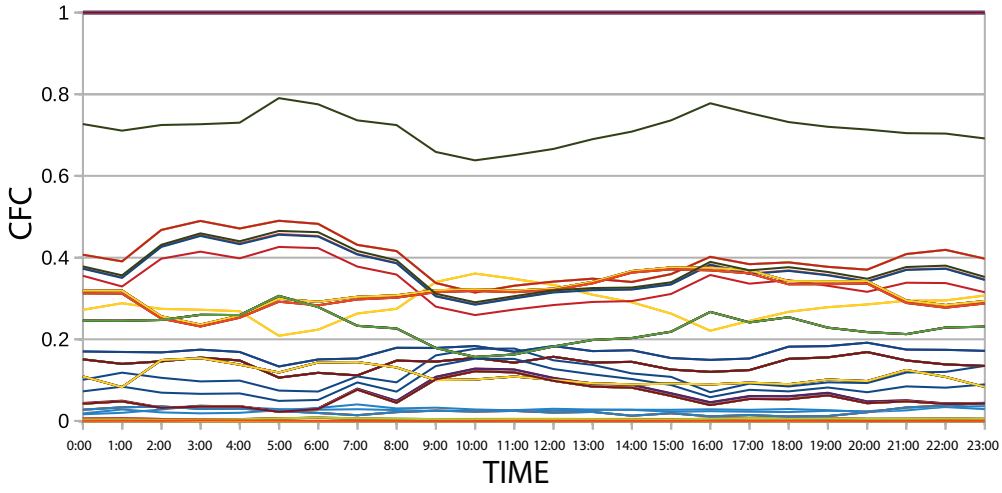


Fig. 8. CFC values for each edge in the electricity network.

To recap, Algorithm 4 results in a set of CFC values $CFC_t(c)$, where t is a timestep and c is a component. For instance, the output for the water system edges can be represented as a matrix CFC^e_{water} in which rows are timesteps and columns are edges:

$$CFC^e_{water} = \begin{vmatrix} CFC_1(e_1) & CFC_1(e_2) & CFC_1(e_3) & \dots & CFC_1(e_{|E|}) \\ CFC_2(e_1) & CFC_2(e_2) & CFC_2(e_3) & \dots & CFC_2(e_{|E|}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CFC_k(e_1) & CFC_k(e_2) & CFC_k(e_3) & \dots & CFC_k(e_{|E|}) \end{vmatrix}$$

Given a matrix of this sort, what method should be used to aggregate the information in the matrix into a single component importance measure for each edge??

The most intuitive approach to ranking the components is to: (1) take the *sample mean* of each column, and; (2) rank columns in descending order. This would be an appropriate strategy if each row of the matrix was a sample from the space of assignments (i.e., in a Monte Carlo approach) at a given time t. However, the rows in the matrix are assessments of the system at different points in time. The use of descriptive statistical measures (e.g., average, variance) elides system dynamics. The same is true of various other methods (e.g., spectral analysis) that might be employed to analyze the matrix.

The choice of ranking approach is dependent upon the purpose of analysis. Consider a long-term (e.g., multi-year) analysis that attempts to study the distribution of critical flow patterns in response to changing population densities and land-use patterns. In such a setting, the limiting behaviour of the system is of interest.

Figure 9 displays a situation in which criticality curves for two different components have the same integral but completely different trends over time. For a long-term (decadal) analysis of infrastructure criticality, the component with the orange criticality curve is clearly the more important of the two. In this setting, some form of trend-based, multi-variable time series analysis is required.

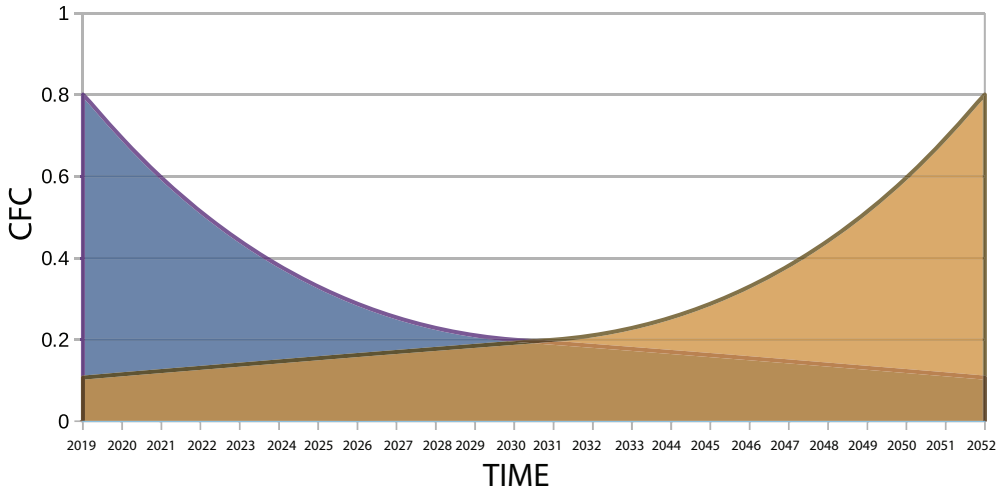


Fig. 9. Two components with similar integrals but different long term behavior.

Since the time series in this paper represent average demands in a *daily cycle*, components are ranked according to the *integral* of their CFC curve. Taking the water network edges as an example, a cubic spline is defined on the set of sample points $\{(CFC_1(e), CFC_2(e), \ldots, CFC_k(e)\}$ corresponding to edge $e \in E$. An integral is calculated from the cubic spline (as shown in **Figure 10**) and normalized by the maximum possible area $MAX\_CFC \cdot (k-1) = 1 \cdot 23 = 23$ (recall that CFC values are already normalized, so that the maximum CFC at any time step is 1). The result is then assigned to the edge e as its *global CFC value* $CFC_G(e)$ for the entire time series. The set of all water network edges E is then ranked by sorting the edges according to their CFC_G values.

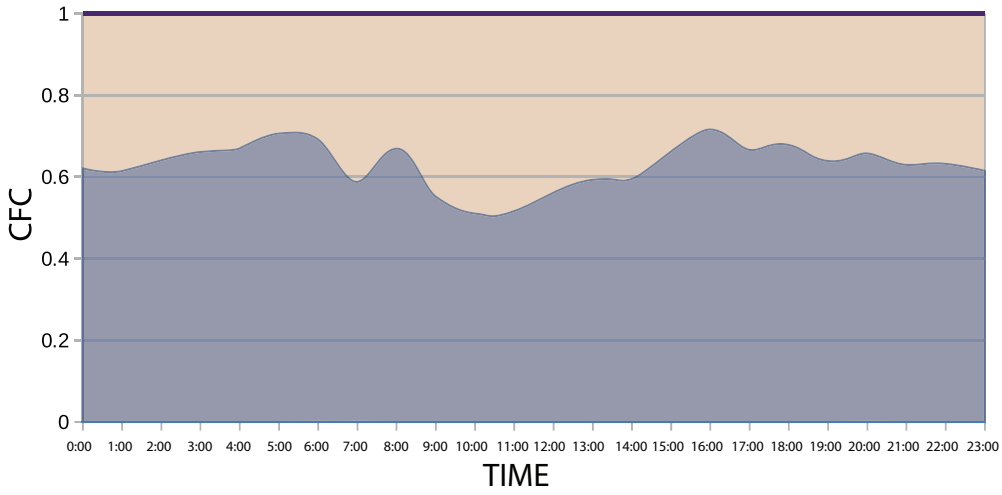


Fig. 10. Computing the global CFC value for a given component e. A cubic spline is overlaid on the CFC values for e. The integral of the spline (blue) is computed and normalized by the total area.

The same process is repeated for vertices, and for the other networks in the system-of-systems. Because of the way in which the critical flow centrality metric is defined, values for edges and vertices are commensurate, allowing a global ranking of all components in the interdependent system.

5 Reliability

5.1 Edge Reliability

In the previous work [53], critical flow complexity values were combined with edge reliability to define a composite measure of component importance. A **reliability function** $r : E \to [0, 1]$ was introduced to assign edges $e \in E$ a reliability rating $r(e) \in [0, 1]$. The **Unreliable Critical Flow** ("UCF") component importance measure was introduced:

$$C'^{UCF}(e) = C'^{CF}(e)(1 - r(e))$$

Note that UCF values lie in the range $[0, 1]$, since the normalized critical flow centrality $C'^{FC}(e)$ and reliability rating $r(e)$ are both in $[0, 1]$.

The use of the UCF measure on interdependent networks with time-varying demands is a straightforward extension of the approach for the CFC. The most notable change is that the reliability rating for a network component e is no longer a single value $r(e) \in [0, 1]$, but rather a time series $R_e = \{r_{e1}, r_{e2}, \ldots, r_{ek}\}$. This allows the modeler to represent different processes (e.g., the deterioration of components over long time periods, or increased risk of damage due to temperature variations within a daily cycle.)

The UCF measures are computed for each timestep t using the CFC values and reliability ratings at t. (Note that these values are scaled at t by dividing each one by the maximum UCF measure for that timestep.) The end result is a matrix in which entry (i, j) gives the UCF values for each edge e_j at timestep i. As in the case of the CFC, a cubic spline is overlaid on the values for each edge, creating an unreliability curve. After computing the integral and dividing it by the maximum possible area, the *global UCF value* for edge e [denoted $UCF_G(e)$] is computed.

Geospatial dependencies between infrastructure components can be introduced into edge reliability analysis in a number of ways. For example, edges that are co-located (e.g., a water pipe and electricity pipe sharing the same service tunnel) could be forced to share the same reliability rating. Co-located components could also be assigned a reliability penalty that reflects the fact that component failures are no longer completely independent.

5.2 Failure Analysis

CFC measures can also be used with another form of reliability analysis in which components are deliberately failed or degraded (e.g., by reducing their capacity) in order to assess the effects on the system as a whole. Recall from the previous work [53] that the **critical flow** in network G given assignment A is the sum of flows reaching the demand nodes, weighted by criticality:

$$F_A^C(G) = \sum_{d \in V_D} f_A(d) c_r(d)$$

The normalized CFC for component e is a measure of the proportion of critical flow passing through e. However, the CFC is relative to an assignment, which includes demand values for all of the demand nodes. A given component e may have a high CFC value under a given assignment, but it may be the case that if e suffers a (partial) failure there are other routes through which flow may travel in order to satisfy critical demand. **Figure 11** provides an illustration of this situation.

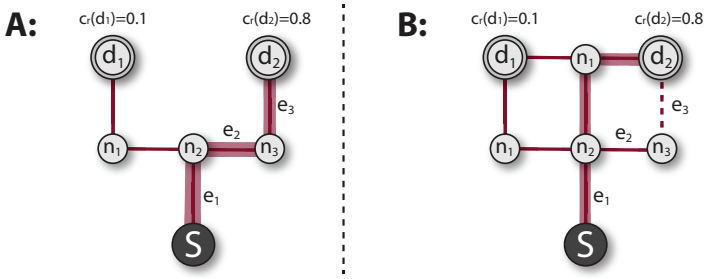


Fig. 11. Two networks with different behaviour in edge failure scenarios. Network A carries most of its critical flow through the path $\{e_1, e_2, e_3\}$. In case of edge failure, no alternative paths are available. Network B has a fallback route in case edge e_3 fails.

As a result, this form of failure analysis provides some indication of whether there are fallback routes that can supply critical flow in the event that a component e fails. If the failure of e consistently results in reduced critical flow across the entire time series, one can assume that e is even more critical than the CFC measure alone might suggest. If large CFC values are correlated with reduced critical flow, then the components that bear the most critical load in the network lack fallbacks.

As a demonstration of the method, **Algorithm 5** shows a high-level view of a procedure in which capacities of edges in a single infrastructure system are degraded one-at-a-time. For each time $t < T$, appropriate demands and criticality values are loaded into the graph. Then each edge $e \in E$ is considered in order, degrading its capacity and performing the CFC computation on the altered network. The critical flow is then used to create a *loss measure* that indicates the amount of critical flow that is lost when edge e is degraded. The **failure loss** $FL_t(e)$ for edge $e \in E$ at time t is:

$$FL_t(e) = 1 - \frac{\sum_{d \in V_D} f_A(d,t) c_r(d,t)}{\sum_{d \in V_D} \delta(d,t) c_r(d,t)}$$

where (recalling Section 3.2) V_D is the set of demand nodes in G, $\delta(d,t)$ is the demand at time t from demand node d, $f_A(d,t)$ is the actual flow to d at time t, and $c_r(d,t) \in [0,1]$ is the criticality rating for d at t.

Function *PerformEdgeFailureAnalysis(S)*
  foreach $t \in [1,T]$ **do**
    LoadDemands(S, t)
    foreach $e \in E$ **do**
      Var originalCapacity ← e.capacity
      e.capacity ← *Degrade*(e.capacity)
      ComputeSingleSystemCFC($\mathcal{S}$)
      ComputeFailureLoss($\mathcal{S}$)
      e.capacity ← originalCapacity
    end
  end

Algorithm 5: Edge failure analysis on network S with time-varying demands.

The edge failure mechanism was tested on the network from **Figure 4** by degrading the capacity of each edge e to 0. (Demands and criticality ratings were the same as in previous sections.) The failure loss $FL_t(e)$ was computed for each edge e at each time $t \in [0, 23]$ and averaged over the 24 hour cycle to create an aggregate failure loss metric. The CFC values for each edge e were likewise averaged over the same time frame.

Figure 12 shows both the averaged CFC and averaged FL metrics for the edges of the water network. Two facts are immediately obvious. First, the vast majority of edges have negligible average CFC and FL values. These are typically low-capacity feeds from a residential street's water pipe to an individual lot/parcel.

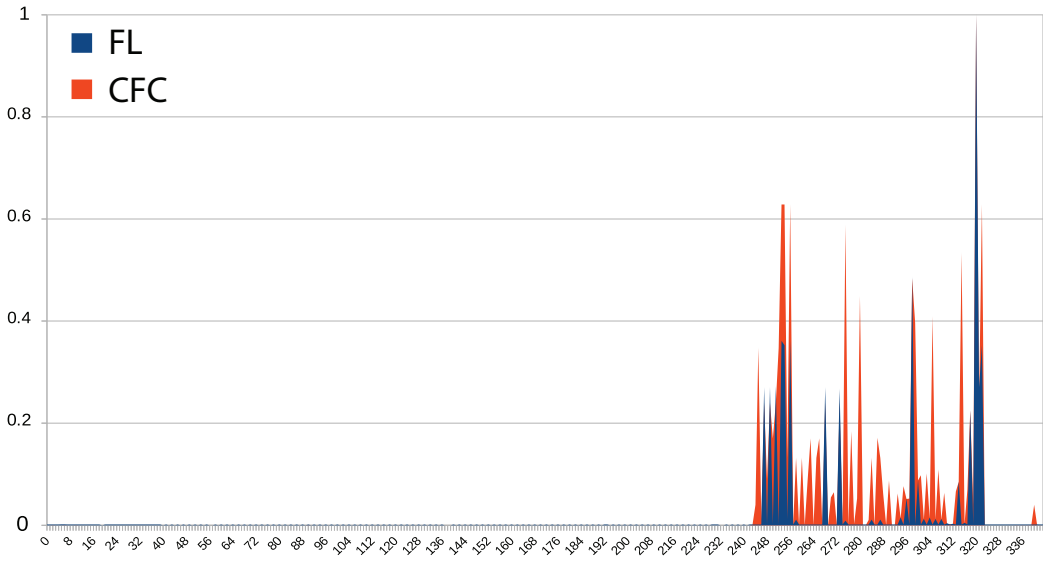


Fig. 12. Averaged *failure loss* (FL) and averaged *critical flow centrality* (CFC) on the edges of the water network in **Figure 4**, computed over a 24-hour period. The majority of edges (e.g., those that feed individual lots) have negligible FL and CFC values.

Second, a significant percentage of of those edges with high CFC ratings also have low FL values. Although these pipe segments carry a sizable amount of critical flow, alternative routes are available in case they should suffer individual failures. Examples include the pipes that define the loops around residential blocks; these loops are resistant to individual failure, since there are two paths from the entry point of the loop to any lot/parcel.

Of course, the main pipes from the reservoir have no backups, as demonstrated by the overlap of FL and CFC values for edge 320. In general, the covariance of FL and CFC is mildly significant but also somewhat misleading as a summary statistic. With a different network topology that included multiple sources and alternative paths, one would expect less correlation between the FL and CFC values, making the easily computable CFC a poor predictor of the consequences of edge failures.

5.3 Limitations and Assumptions

The edge failure analysis presented above was subject to several simplifying assumptions. First, *geospatial dependencies* were not included in the analysis. This decision was made in the interests of brevity, but a thorough treatment of edge failure in an interdependent network would study simultaneous failure of co-located components.

Second, the definition of the FL metric uses the aggregate of all demands at the network's demand nodes as the normalizing factor. This is appropriate for a network where all demands are satisfied in the baseline state, but it will overestimate losses in networks which exhibit unsatisfied demand. For the scenario under study (i.e., daily critical flow distribution in a neighborhood), this assumption is reasonable.

Third, physical (resource) dependencies were not accounted for. The failure analysis method focused on a single infrastructure system (e.g., a water network). However, the failure/degradation of electrical components would cascade into the water network, since the latter requires electricity to operate water pumps.

The work in this paper (as well as its predecessor [53]) assumes that the modeling and solution methods used in each layer are not necessarily commensurate. That is, the electricity layer may be modeled with one set of domain-specific techniques, while the water layer may be modeled with another. All that is required is for each layer to provide flow computation and a basic network topology. This decision, while realistic from a software engineering perspective, precludes the use of standard approaches to modeling cascading failures.

To model physical dependencies and cascading failure in such a setting requires the use of additional machinery. Component failure in the electricity system could result in reduced power levels at the water pumps; this, in turn, could alter water distribution flows, result in reduced electricity demand from other components of the water system — thereby changing demand patterns for the electricity system. Thus, the result is an equilibrium problem in which changes in one layer percolate through other layers, and then back again. Solving such a problem is well beyond the scope of this paper.

6 Conclusion

This paper extended previous work [53] on *critical flow centrality* ("CFC") from independent networks with integer-valued demands to interdependent networks with time-varying demands. It provided a mechanism for representing dependencies between layers of a complex infrastructure network, as well as a discussion of how CFC values may be computed over time series. Lastly, the paper discussed how CFC computations may be integrated with two forms of *reliability analysis* — edge reliability metrics and component failure/degradation models.

Connections between individual infrastructure systems are modeled through the use of geospatial and physical (i.e., resource-based) dependencies. Both types of dependency can be used to define a graph $\mathcal{G}$ in which nodes are individual infrastructure systems and edges appear when dependencies are present. In the case of physical dependencies, it was assumed that $\mathcal{G}$ is a directed, acyclic graph that can be ordered with a topological sort.

Similarly, the time series computation presented in the paper assumed that the time-varying demands represented average values in a 24-hour cycle. This decision simplified the analysis, and allowed the use of integrals to compute a global CFC value for the

entire cycle. For the study of trends in infrastructure systems over time, the integral-based aggregation would need to be supplanted by trend-based, multi-variable time series analysis.

In general, the methods presented in the paper only work for medium/long time scales. The main culprit is the use of integer-valued representations for demands and capacity constraints. This decision, which was made in order to simplify the problem and avoid numerical instability, means that short term dynamics are impossible to represent. This precludes forms of analysis in which the rates of change (e.g., of flow) on system components may be analyzed.

Many avenues of future work remain, the most important of which is removing the restriction of $\mathcal{G}$ to directed, acyclic graphs. To do so invites consideration of equilibrium concerns — changes in one network cause changes in others, altering flow distributions and demand patterns in complex ways. Providing solutions for this type of problem is well outside the scope of the present paper.

Even with this restriction in place, there are still additional issues to be resolved. First, a more realistic flow mechanism (e.g., domain-specific methods) should replace the generic Edmonds-Karp algorithm that favors shortest paths (thereby introducing artifacts into the flow solution). Second, geospatial dependencies should be introduced into both the edge reliability and component failure analyses. Additional avenues of future research were hinted at throughout the paper.

7 Acknowledgements

Funding for this work was provided by an Ontario Research Fund — Research Excellence Round 7 grant for the "iCity: Urban Informatics for Sustainable Metropolitan Growth" project. The author wishes to Eric Miller, Mark Fox, Steve Easterbrook, and the rest of the researchers involved with the iCity project for the opportunity to work on problems outside of his existing areas of expertise.

References

[1] Ravindra Ahuja, Thomas Magnanti, and James Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ, 1993.

[2] Massoud Amin. Toward Secure and Resilient Interdependent Infrastructures. *Journal of Infrastructure Systems*, 8(3):67–75, 2007.

[3] Hadi Amini, Kianoosh G. Boroojeni, S S Iyengar, Panos M Pardalos, Frede Blaabjerg, and Asad M Madni. *Sustainable Interdependent Networks: From Theory to Application*. Springer, 2018.

[4] M. Hadi Amini, Kianoosh G. Boroojeni, S. S. Iyengar, Panos M. Pardalos, Frede Blaabjerg, and Asad M. Madni. *Sustainable Interdependent Networks II: From Smart Power Grids to Intelligent Transportation Networks*. Springer, 2019.

[5] George E Apostolakis and Douglas M Lemon. A screening methodology for the identification and ranking of infrastructure vulnerabilities due to terrorism. *Risk Analysis*, 25(2):361–376, 2005.

[6] Aquacraft. Embedded Energy in Water Studies, Study 3: End-use Water Demand Profiles. Technical report, California Public Utilities Commission, Energy Division, 2011.

[7] Robert F. Austin, David P. DiSera, and Talbot J. Brooks. *GIS for critical infrastructure protection*. CRC Press, 2016.

[8] Alessandro Birolini. *Reliability Engineering Theory and Practice*. Springer, Berlin Heidelberg, 5 edition, 2007.

[9] S Boccaletti, G Bianconi, R Criado, and C I Genio. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[10] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, 2010.

[11] Sanjay K. Chaturvedi. *Network Reliability: Measures and Evaluation*. Scrivener Publishing, Hoboken, New Jersey, 2016.

[12] Ali A. Chowdhury and Don O. Koval. *Power Distribution System Reliability: Practical Methods and Applications*. Wiley, Hoboken, New Jersey, 2009.

[13] Robert M. Clark, Simon Hakim, and Avi Ostfeld. *Handbook of Water and Wastewater Systems Protection*. Springer, 2011.

[14] Thomas Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 3 edition, 2009.

[15] National Research Council. Drinking Water Distribution Systems: Assessing and Reducing Risks. Technical report, Committee on Public Water Supply Distribution Systems: Assessing and Reducing Risks, 2006.

[16] Gregorio D'Agostino and Antonio Scala. *Networks of Networks: The Last Frontier of Complexity*. Springer, 2014.

[17] L. Dall'Asta, A. Barrat, M. Barthélemy, and A. Vespignani. Vulnerability of weighted networks. *Journal of Statistical Mechanics: Theory and Experiment*, page P04006, 2006.

[18] L. Duenas-Osorio, J.I. Craig, B.J. Goodno, and A. Bostrom. Interdependent response of networked systems. *Journal of Infrastructure Systems*, 13(3):185–194, 2007.

[19] L.C Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41, 1977.

[20] Giulio Galvan and J Agarwal. Vulnerability analysis of interdependent infrastructure systems. In T Haukaas, editor, *Proceedings of the 12th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP12): Vancouver, Canada, July 12-15 University of British Columbia.*, 2015.

[21] Antonios Garas. *Interconnected networks*. Springer, 2016.

[22] Ilya Gertsbakh and Yoseph Shpungin. *Network Reliability and Resilience*. Springer, 2011.

[23] Michael Krivelevich, Choongbum Lee, and Benny Sudakov. Resilient pancyclicity of random and pseudo-random graphs. *SIAM Journal on Discrete Mathematics*, 24:1–17, 2009.

[24] Wolfgang Kröger and Cen Nan. Addressing Interdependencies of Complex Technical Networks. In Gregorio D'Agostino and Antonio Scala, editors, *Networks of Networks: The Last Frontier of Complexity*. Springer, 2014.

[25] Vito Latora and Massimo Marchiori. Efficient Behavior of Small-World Networks. *Physical Review Letters*, 87(19):3–6, 2001.

[26] Earl E. Lee, John E. Mitchell, and William A. Wallace. Restoration of services in interdependent infrastructure systems: A network flows approach. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 37(6):1303–1317, 2007.

[27] Ted G. Lewis. *Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation*. Wiley-Interscience, 2006.

[28] Javier Lopez, Roberto Setola, and Stephen D. Wolthusen. *Critical Infrastructure Protection: Information Infrastructure Models, Analysis, and Defense*. Springer, 2012.

[29] M. Mair, R. Sitzenfrei, M Moderl, and W Rauch. Identifying multi-utility network similarities. In *World Environmental and Water Resources Congress 2012: Crossing Boundaries*, pages 3147–3153, 2012.

[30] Michael Mair, Jonatan Zischg, Wolfgang Rauch, and Robert Sitzenfrei. Where to find water pipes and sewers? - on the correlation of infrastructure networks in the urban environment. *Water*, 9(146), 2017.

[31] Didrik Meijer, Marco van Bijnen, Jeroen Langeveld, Hans Korving, Johan Post, and François Clemens. Identifying critical elements in sewer networks using graph-theory. *Water*, 10(2), 2018.

[32] Mark Newman. *Networks: An Introduction*. Oxford University Press, 2 edition, 2018.

[33] Albert Nieuwenhuijs, Eric Luiijf, and Marieke Klaver. Modeling Dependencies in Critical Infrastructures. In Mauricio Papa and Sujeet Shenoi, editors, *Critical Infrastructure Protection II*, pages 205–213. Springer, 2008.

[34] Paul Novak, Vincent Guinot, Alan Jeffrey, and Dominic E Reeve. *Hydraulic Modelling – an Introduction: Principles, Methods and Applications*. Spon Press, 2010.

[35] Min Ouyang, Lijing Zhao, Liu Hong, and Zhezhe Pan. Comparisons of complex network based models and real train flow model to analyze Chinese railway vulnerability. *Reliability Engineering and System Safety*, 123:38–46, 2014.

[36] Mauricio Papa and Sujeet Shenoi. *Critical Infrastructure Protection II*. Springer, 2008.

[37] James P. Peerenboom and Ronald E. Fisher. Analyzing cross-sector interdependencies. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 1–9, 2007.

[38] Per Hokstad, Ingrid Bouwer Utne, and Jorn Vatn. *Risk and Interdependencies in Critical Infrastructures: A Guideline for Analysis*. Springer, 2012.

[39] Frédéric Petit, Duane Verner, Julia Phillips, and Lawrence Paul Lewis. Critical Infrastructure Protection and Resilience—Integrating Interdependencies. In A. J. Masys, editor, *Security by Design*. Springer, 2018.

[40] Mauricio G.C Resende and Panos M Pardalos. *Handbook of Optimization in Telecommunications*. Springer, New York, NY, 2006.

[41] Steven M. Rinaldi, James P. Peerenboom, and Terrence K. Kelly. Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems Magazine*, 21(6):11–25, 2001.

[42] Roberto Setola, Sandro Bologna, Emiliano Casalicchio, and Vincenzo Masucci. An Integrated Approach for Simulating Interdependencies. In Mauricio Papa and Sujeet Shenoi, editors, *Critical Infrastructure Protection II*. Springer, 2008.

[43] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications*. Springer, 2017.

[44] Benny Sudakov and V.H. Vu. Local resilience of graphs. *Random Structures and Algorithms*, 33(4):409–433, 2008.

[45] Nils K. Svendsen and Stephen D. Wolthusen. Analysis and statistical properties of critical infrastructure interdependency multiflow models. *Proceedings of the 2007 IEEE Workshop on Information Assurance, IAW*, pages 247–254, 2007.

[46] Nils Kalstad Svendsen and Stephen D. Wolthusen. Connectivity models of interdependency in mixed-type critical infrastructure networks. *Information Security Technical Report*, 12(1):44–55, 2007.

[47] Nils Kalstad Svendsen and Stephen D. Wolthusen. Graph models of critical infrastructure interdependencies. *Lecture Notes in Computer Science*, 4543 LNCS:208–211, 2007.

[48] Nils Kalstad Svendsen and Stephen D. Wolthusen. An analysis of cyclical interdependencies in critical infrastructures. *Lecture Notes in Computer Science*, 5141 LNCS:25–36, 2008.

[49] Michael A P Taylor. *Vulnerability Analysis for Transportation Networks*. Elsevier, Amsterdam, Netherlands, 2017.

[50] My T Thai and Panos M Pardalos. *Handbook of Optimization in Complex Networks: Theory and Applications*, volume 57. Springer, 2012.

[51] Alessandro Vespignani. Complex networks: The fragility of interdependency. *Nature*, 464(7291):984–985, 2010.

[52] Robert S Wilkov. Analysis and. Design of Reliable Computer Networks. *IEEE Transactions on Communications*, 20(3):660–678, 1972.

[53] James B Williams. Identifying Sensitive Components in Infrastructure Networks via Critical Flows. *Journal of Infrastructure Systems*, XXX(XXX):XXX–XXX, 2019.

[54] K Wolter, A Avritzer, and M Vieira. *Resilience assessment and evaluation of computing systems*. Springer, 2012.

[55] Rae Zimmerman and Carlos E Restrepo. The next step : quantifying infrastructure interdependencies to improve security. *Int. J. Critical Infrastructures*, 2:215–230, 2006.

[56] E. Zio and G. Sansavini. Modeling failure cascades in critical infrastructures with physically-characterized components and interdependencies. In *ESREL 2010 Annual Conference*, pages 652–661, 2010.