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

James Bryan Williams

University of Toronto, Department of Computer Science, Canada

Abstract

This paper presents a method that allows urban planners and municipal engineers to identify critical components of interdependent infrastructure systems. The intent of the method is to provide a means of modeling the impact of capacity-related changes (e.g., population growth, component degradation) on a city's ability to deliver resources to critical locations. Infrastructure systems are modeled as flow networks in which capacities, demands, and supply constraints vary over time; demand nodes also have criticality ratings that allow a user to model levels of importance. Interconnections between infrastructure systems are represented by physical and geospatial dependencies at a component level. A flow-based centrality measure is used to rank components according to their role in the delivery of resources to critical locations. A simple instantiation of the method is presented and evaluated on a district-scale model of a city that contains interconnected water and electricity networks. Finally, two forms of reliability analysis are demonstrated: a composite measure incorporating edge reliability, and a variation on standard component failure/degradation analysis.

Keywords: Component importance measures, Centrality measures, Complex systems, Network science, Infrastructure reliability

1. Introduction

This paper presents a method for identifying critical components in interdependent, urban infrastructure systems, where a component is deemed critical to the extent that it is relied upon to deliver resources to critical locations (e.g,. hospitals). The goal of the work is to provide urban planners and municipal engineers with a means of visualizing the impact of capacity-related changes. This goal is met by combining a flow-based centrality measure with a network-based approach to infrastructure systems that permits modelers to: (1) define physical and geospatial interdependencies; (2) represent temporal dynamics through the use of time series for key system variables, and; (3) label locations with criticality values in order to represent asset importance. While useful in its own right, this basic method may be combined with standard reliability methods in order to model the effects of component failure.

The perspective in this work is resource-based. 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 resources and/or services can have extremely deleterious consequences, particularly for critical locations such as hospitals and transportation hubs. Methods for identifying infrastructure components that are relied upon to supply critical locations with resources can be used for a variety of purposes, including maintenance scheduling, disaster recovery, critical infrastructure protection, and urban planning.

In keeping with the focus on resource delivery, the method presented in this work represents infrastructure systems as *flow networks*–graphs that allow commodities to flow from a set of supply nodes to a set of demand nodes. Edges of the network (i.e., pipes) are labeled with capacity values that limit their maximum flow. Demand nodes represent consumers of resources such as hospitals, laundromats, and homes. The modeler supplies the various system variables, such as capacities, supply constraints, and demands.

Unlike typical flow networks, however, the demand nodes are labeled with *criticality ratings* that indicate the relative importance of the structure at that location. These criticality ratings, which are also defined by the modeler, are then used to calculate a centrality measure–the *critical flow centrality* ("CFC"). CFC values are propagated from the demand nodes through the rest of the infrastructure system's components. The user can then visualize the system's components (e.g., with a heat map) according to their role in delivering resources to critical locations.

Since urban infrastructure systems are interconnected in various ways, users may define physical and geospatial *dependencies*. Physical dependencies exist when components of system A require resources supplied by system B. Geospatial dependencies exist when components of A and B are co-located. The inclusion of dependencies allows the user to model the potential for cascading failure, simultaneous disruption due to weather events, and other common scenarios. In particular, criticality ratings are propagated across interdependent systems so that the user may immediately determine which components of System A play an important role in supplying the critical components of System B with resources.

Apart from dependency modeling and criticality ratings, the third distinguishing characteristic of this work is the use of *time series*. Key system variables (e.g., capacity constraints, supply constraints, demand levels, and criticality ratings) are dynamic rather than static, allowing the user to model phenomena at a variety of time scales. For instance, a geographer may be interested in studying the impact of population growth on a particular district over decades, while an urban planner might use the same model with different data in order to visualize the short-term consequences of planning interventions.

Finally, the CFC measure becomes even more interesting when combined with standard reliability methods. The CFC was designed to identify components that play a substantial role in delivering resources to critical locations. However, the failure of a component with a high CFC value will not necessarily result in a loss of service, as there may be alternative (fallback) routes available. Combining the CFC with reliability methods allows a modeler to reason explicitly about fallback routes and the consequences of component failure.

The structure of this paper is as follows. Section 2 provides useful background information, while Section 3 introduces the methodology used in this paper, including the CFC measure. Section 4 provides an evaluation of the methodology on a district-scale model of a city. Section 5 discusses two forms of reliability analysis that can be combined with the CFC measure to model potential service loss due to component failure. The paper closes with suggestions for future research.

2. Background

The method in this paper can be viewed as a combination of techniques from network science and critical infrastructure protection. The fundamental building block is a *component importance measure* ("CIM") (e.g,. [1], [2], [3]) that estimates the degree to which a given component participates in the delivery of resources to critical locations. Before discussing the method in detail, a quick discussion of relevant background material is required.

2.1. Network Science and Centrality Measures

Networks are a common choice of modeling mechanism in many fields, and critical infrastructure protection is no exception (see [4]). For example, many approaches to infrastructure vulnerability and resilience make use of techniques from network science. From the perspective of the current paper, the most important of these techniques are the *centrality measures*, which are used to identify the most central components in a network (see [5, 6, 7]).

Numerous centrality measures exist [8], the most intuitive of which are: (1) *nearness measures*, which determine a given component's centrality by means of its proximity to other components, and; (2) *betweenness measures*, which deem components to be central to the extent to which they stand between other components as intermediaries. While nearness and betweenness measures focus on network topology, *dynamical measures* are based on processes (e.g., flows, diffusion) taking place on the network.

The progenitor of the method used in this paper is *flow centrality* [9]. Consider a simple network with nodes V and links E. A node v is considered to be *between* other nodes u and w to the extent that the maximum flow between u and w depends on v. Nodes are deemed *central* to the extent that they facilitate maximum flow.

Stated formally, for $u, v, w \in V$, let $m_{u,w}$ be the maximum flow between u and w, and let $m_{u,w}(v)$ be the maximum flow between u and w that depends on v. Then the **flow centrality** ("FC") of a node $v \in V$ is the degree to which the maximum flow between all unordered pairs of nodes depends on v:

$$C^F(v) = \sum_{u \neq w \neq v} m_{u,w}(v) \tag{1}$$

2.2. Interdependent Infrastructures

Infrastructure systems are typically coupled to the extent that the failure of components in one system can cause failures in connected systems [10]. These *interdependent* systems are typically more fragile than solitary systems [11], with additional failure modes (e.g., *cascading failures* [12]) that can be quite complex. For example, water distribution systems impose much greater cascading damage on other systems than they receive in return [13], and they seem to display a greater propensity to initiate cascading failure in other systems [14].

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 [4, 15, 16]. Component failure has been studied extensively in the field of reliability engineering (e.g., [17]) and in the various engineering disciplines (e.g., water [18], drainage [19], electricity [20], telecommunications [21], and transportation [22]). Disruption of networks has also been considered in operations research (e.g., [23]), computer science (e.g., [24, 25]), network reliability (e.g., [26, 27]), graph theory (e.g., [28, 29]), and network science (e.g., [30]).

However, disruptions are not the only phenomena to have an impact on urban infrastructure. Infrastructure systems are influenced by a variety of factors, including: (1) *population growth*, which can push the capacity constraints of legacy infrastructure; (2) *component degradation*, which can reduce component capacity; (3) *maintenance activities*, which can interdict flow through selected components, and; (4) *planning interventions* (e.g., the development of new residential subdivisions), which can have effects both on system topology and on demand patterns. In some cases, these factors can alter flow distributions so that resources needed by critical locations are dependent on unreliable infrastructure.

Various research communities have advocated an integrated view of infrastructure systems, and a growing body of work is available on interdependencies (e.g., [31, 32]). For instance, homeland security initiatives following the September 11th terrorist attacks in the United States spurred numerous efforts addressing infrastructure interdependencies (e.g., [33]). Overviews of techniques for the modeling and simulating interdependent critical infrastructure systems may be found in several places, including [34].

2.3. Modeling Interdependent Infrastructures with Networks

One approach to analyzing interdependent infrastructure systems involves modeling them as *interdependent networks* [32]. Interdependent (or *multilayer* [35]) networks have received increasing amounts of attention of late, particularly from the physics and network science communities. A recent survey paper can be found in [36], while books on the topic are readily available (e.g., [37, 38, 32, 39, 40, 35]).

Network A is **dependent** on network B if the state of B can influence the state of A [41] (see also [42]). Dependencies can be classified as follows [14]:[1]

1. **Physical dependencies**, in which the state of A is affected by the material outputs/flows of B;
2. **Geospatial dependencies**, in which certain components of A and B are in such close spatial proximity such that local events can affect both networks;
3. **Informational dependencies**, in which A and B are connected by information and communications technology;
4. **Social dependencies**, in which A affects B along social dimensions;
5. **Procedural dependencies**, where A affects B on the basis of organizational or regulatory structures, and;
6. **Financial dependencies**, where market conditions, financial crises and other economic events allow one network to affect another.

[1] Alternative classifications appear in [43, 44].

There are many ways to represent these dependencies in network models, a discussion of which is beyond the scope of the paper.

2.4. 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 [45] 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 [20]). 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. [43] 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. [46] 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 [47]; (3) clustering coefficient [48], 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 [49] 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 [50, 51, 52, 53]. 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.

Kotzanikolaou et al [54] provide a method for identifying threats to infrastructure systems that may have a significant cumulative effect. From a risk analysis table detailing the risks to particular infrastructures, the authors construct a *risk dependency graph* ("RDG") in which: (1) a node represents an infrastructure system, and; (2) a directed edge $X \rightarrow Y$ from system X to system Y represents a risk dependency (i.e., X poses a risk to Y). Edges are labeled with risk values that represent the likelihood of disruption, as well as societal impact. The authors provide a method for computing higher order risks from such a graph.

Stergiopoulos et al [55] use an RDG in combination with centrality measures (e.g., betweenness, eigenvector centrality, node degree) to identify critical systems. The centrality measures are computed in order to identify nodes that affect critical risk paths in the RDG. A decision-tree algorithm is then used to select a subset of these nodes for risk mitigation. Testing their approach on empirical data, the authors make a number of observations about the relationship between centrality and risk (e.g., that the most critical paths in the RDG tend to involve nodes with high centrality).

Shahraeini and Kotzanikolaou [56] provide a method to aid in the design of *wide area measurement systems* ("WAMS"), which are composed of the measuring and communications layers of a smart grid. The authors provide a model that captures the internal dependencies between these layers in a dependency graph. Given such a graph, the importance matrix of the bus components is determined, and then the importance metrics of all of the rest of the WAMS components are computed using a centrality measure. An optimization algorithm is used to distribute the importance of the WAMS elements in order to avoid single points of failure. Interestingly, while the authors acknowledge that the importance of power grid components is dynamic, changing over time, they use a non-temporal approach for the initial design of a WAMS.

The present work is different from these examples of prior art. As discussed below, the entire set of interdependent systems is represented as a graph. Similarly to RDGs, each node is an individual infrastructure system. However, unlike RDGs, the edges in the graph do not represent risk dependencies; rather, they represent actual component-level interdependencies. Furthermore, the graph formed from the interdependent systems is used only for controlling the order of execution, and not for analysis.

Of all the aforementioned works, the method in this paper is closest to Apostolakis and Lemon [45] in overall intent. In both cases the computational model allows for co-location. However, the present work also allows for physical dependencies, as well as dynamic behavior through the use of time series for key system variables. The temporal aspects of the current work also distinguish it from many of its predecessors.

3. Methodology

The goal of this work is to explore means by which urban planners, municipal engineers and other decision makers can identify critical components of interdependent infrastructure networks. When embodied in software, such methods can be used to support decision makers engaged in maintenance scheduling, zoning, capacity planning, or other activities related to municipal infrastructure.

3.1. Overview

The paper provides an example of such a method, based on a centrality measure that combines classical flow centrality [9] with concepts from critical infrastructure systems (e.g., [45]). The perspective in the paper is *resource-based*, focusing on the routes by which resources are delivered to consumers. Components are deemed critical to the extent that they are involved in facilitating the flow of resources to critical locations.

Computation of the centrality measure, *critical flow centrality* ("CFC"), can be accomplished in several ways (see [57]). In the current paper, a discrete-valued approach is taken in which: (1) an infrastructure system is represented as a flow-network; (2) demands, capacities, and supply limits are given as integers, and; (3) each demand node in the network is assigned a real-valued criticality rating. Network flows are simulated with a standard maximum flow algorithm; once a flow has been defined, a search algorithm computes expected contribution of each component to the critical flow within the network.

Since infrastructure networks are not independent of each other, physical and geospatial dependencies may be introduced between individual infrastructures. The most important of these for the present paper are *physical dependencies* in which resources provided by one system (e.g., electricity) are used by another system (e.g., water pumps). One of the main contributions of the paper is to show how CFC values can be propagated from one infrastructure system to another.

The method is demonstrated by applying it to a district-level model of a city. Each structure has a type, a criticality rating, and a set of demand curves (time series) for resources. For reasons of brevity, only two infrastructure systems (electricity, water) are shown. The simple method provided in this paper also assumes that the physical dependencies between individual infrastructures are acyclic.

The main thrust of the demonstration is to show that: (1) the computation of CFC values can be performed efficiently, enabling their use in interactive GIS applications; (2) CFC values can correctly propagate between system models, and; (3) CFC computations can be integrated with standard reliability measures to provide a composite view of a system.

The CFC measure itself is completely general, requiring only a flow solution and a network topology. The method presented in this paper uses the discrete algorithms to compute values for each individual infrastructure system—namely, (1) an integer-valued, maximum-flow algorithm to approximate resource flow within infrastructures, and; (2) a modified graph-search algorithm to compute CFC values. These design choices are for ease of explanation, and more sophisticated, heterogeneous systems can be accommodated. One can model a water system using hydraulic techniques [58], for example, coupling it to an electricity system that is simulated using its own domain-specific methods. Given a flow solution and network topology, CFC values can be computed by using Markov-chain Monte Carlo or random walks (see [57] for details).

3.1.1. Integration with GIS

This work was motivated by the problem of providing adequate decision support for urban planning. For instance, densification of urban areas is accompanied by greater demand for resources; the increased demand could: (1) violate capacity constraints, as in the case of the London sewer systems [59, 60], or; (2) threaten the ability of a legacy infrastructure system to reliably deliver services to critical locations such as hospitals and transportation hubs. Urban planners could benefit from tools that allow them to visualize the impacts of land-use decisions on the provision of critical resources and/or services.

Effective modeling of integrated infrastructure systems requires more than a static, single-perspective approach. Management of disruption (and prevention of cascading failures) requires an understanding of system dynamics [61]. Furthermore, any model used to study the disruption of interdependent infrastructures needs to support two different perspectives [43]: (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.

One means of providing infrastructure models that support multiple perspectives is through the use of *geographical information systems* ("GIS") software. In fact, the critical information protection community has begun to use GIS as a platform for resilience and vulnerability analysis [62]. For this reason, the method described in this paper was explicitly designed for integration within GIS software.

3.1.2. Data Sources

Two major challenges arise when data sources are considered. First, data on infrastructure systems does not always exist, and particularly not in a form that permits detailed analysis of interdependencies. Second, infrastructure systems in many countries (e.g., the United States power grid) are not under the control of a single entity [61], making the data collection process difficult. The lack of information on infrastructure assets has motivated some researchers to develop techniques for inferring asset locations from proxy data sources (e.g., [63, 64]).

The model used in this paper is a mixture of synthetic and empirical components. The basic topology (i.e., road and parcel structure) was taken from downtown Toronto, albeit the boundaries were simplified in order to make diagrams feasible and to convey the basic method clearly. Resource demand data was obtained from published studies (e.g., [65]) and from municipal utilities.

3.1.3. Implementation

The sample method was implemented directly in C++ and OpenGL. Road and building information was obtained from OpenStreetMaps, imported into ESRI CityEngine, and edited manually to remove artifacts. Custom python scripts were used to export the road network topology, block/lot geometry, and building shapes from CityEngine to *Extensible Markup Language* ("XML") files. Infrastructure systems were created manually using the application's editing functionality. Lastly, the diagrams shown in this paper were generated by exporting model geometry directly to *Scalable Vector Graphics* ("SVG") format.

3.2. Modeling Approach

This section discusses the building blocks of the simplified model, including: (1) the network representation; (2) the use of time series for key system variables; (3) criticality ratings, and; (4) interdependencies.

3.2.1. Network Representation

An infrastructure system is modeled 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:

- 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 $\|\vec{w}(v_i) - \vec{w}(v_j)\|_2$.

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.

As a flow 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 $V_D = \{d_1, d_2, \ldots d_k\} \subseteq V$ with $V_S \cap V_D = \varnothing$. All other nodes are called *transmission nodes*. Multi-functional nodes are supported using a standard maximum flow reduction (as described in Section 3.4.1).

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 $\sum_{v \in V} f((v_t, v)) = 0$.

A given network G supports a single type of resource/commodity, unlike the multi-commodity approach in [52]. The **value of a flow** is defined as the flow exiting the source nodes: $|f| = \sum_{v \in V} \sum_{s \in S} f(s, v)$.

3.2.2. Supply Constraints and Demand Distributions

Supply constraints and resource demands are represented as discrete, integer-valued *time series* (see [66]). (While capacities can also be represented as time series, the demonstration assumes node and edge capacities are static.) For simplicity, each time series is assumed to be regularly sampled at times $t_i \in T = [0, \infty]$. They can be interpreted 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 functions (time series) for all relevant nodes. 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.

3.2.3. Criticality Ratings

A **criticality function** $cr : V_D \rightarrow \mathbb{R}$ maps demand nodes $d \in V_D$ to a **criticality rating** $cr(d)$. Although it is possible to use *binary* (e.g., critical, non-critical) or *categorical* (e.g., low, medium, high) representations, this paper focuses on the *continuous* variant in which criticality ratings take on values between 0 and 1.

3.2.4. 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.
2. **physical dependencies**, wherein elements in network A require resources flowing through network B.

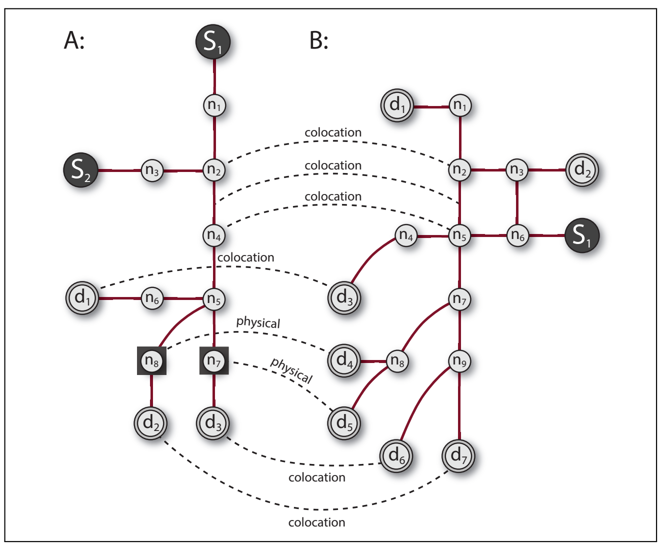

Fig. 1. Infrastructure systems (A) and (B), linked by geospatial and physical dependencies.

Dependencies are represented as *interlinks* between networks [35]. In contrast to Apostolakis and Lemon [45], nodes from one network S_i do not appear directly in another network S_j. This design choice makes it easier to integrate disparate modeling methods for each individual infrastructure system (see [67]).

Interlinks representing physical dependencies are implemented with the use of *interconnection records*. 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** $IR(v_1, v_2)$. 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 $f^R : S_1.V \longrightarrow \mathbb{R}$. 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.

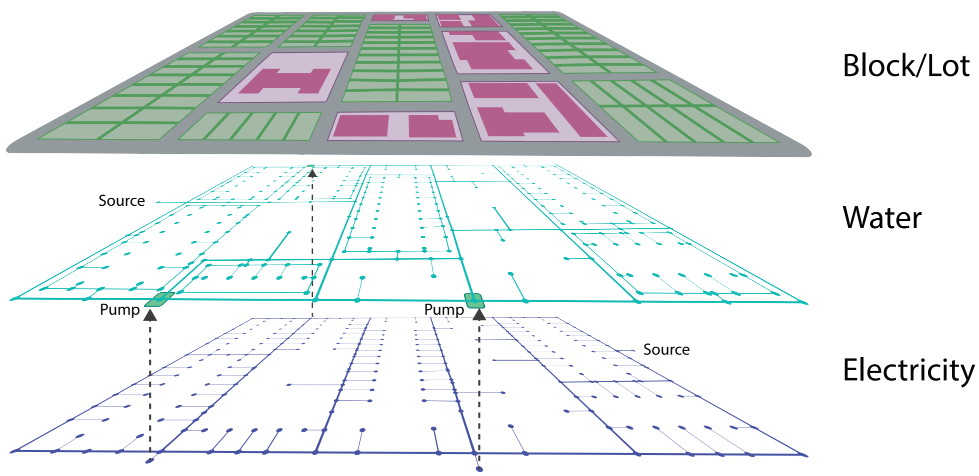

Fig. 2. Infrastructure model with two layers, showing resource flow between water pumps and electricity nodes.

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 $S_1 \longrightarrow S_2$. Mutual dependency between systems makes the computational task more difficult. The methods of Svendsen and Wolthusen (e.g., [52]) accommodate mutual dependencies using multi-commodity flows, but 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 [68]). In contrast, geospatial dependencies are not restricted in such a fashion.

3.3. Critical Flow Centrality

The **Critical Flow Centrality** (“CFC”) of a component is a measure of its role in supplying resources to critical locations [57]. Recall that the **flow** in network G (given assignment A) is the aggregate of all flows reaching the demand nodes:

$$F_A(G) = \sum_{d \in D} f_A(d) \tag{2}$$

The **critical flow** in network G (given assignment A) is the set of flows reaching the demand nodes, weighted by criticality:

$$F_A^C(G) = \sum_{d \in D} f_A(d) c_r(d) \tag{3}$$

A component c (i.e., node or edge) is deemed to be important to the extent that it carries critical flow. Let $f_A(c, d)$ be the flow that reaches $d \in D$ from c given assignment A, and let $E[f_A(c, d)]$ be its expectation. Then the **critical flow centrality** (“CFC") of component c under assignment A is:

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

This quantity may be normalized by the critical flow $F_A^C(G)$:

$$C'^{CF}(c) = \frac{C^{CF}(c)}{F_A^C(G)} = \frac{\sum_{d \in D} c_r(d) E[f_A(c, d)]}{\sum_{d \in D} c_r(d) f_A(d)} \tag{4}$$

Computing the CFC thus reduces to computing the probability $p(d|c)$ that a unit of commodity passing through component c ends up in demand node d. While there are numerous ways to accomplish this task (e.g., Markov chains), this paper uses a discrete, search-based approach.

For each time step t, a flow solution $F(t)$ is generated and represented in a **secondary graph** G' that gives an adjacency-list representation of the stochastic transition matrix induced by $F(t)$. All nodes of G are present in G', but edges of G are only present if they have non-zero flow under $F(t)$. Each node in G' has an *outgoing edge list* that lists the probability that a unit of flow travels down an outgoing edge. Finally, each component (i.e., edge or non-demand node) c in G' has a *map* that lists the demand nodes reachable from c in G'. An entry in this map contains a tuple $\langle d, P(d|c) \rangle$ that gives the probability that a unit of flow passing through c reaches demand node d. **Equation 4** can be computed from these maps.

The CFC calculation proceeds by performing a reverse *depth-first search* ("DFS") on G' for each demand node d. When invoked, the DFS computes the probability that each edge or non-demand node in G' sends flow to d. A given node or edge (identified by an *ID* number) may be visited multiple times in the course of the search, requiring care to avoid pushing superfluous probability. (This method does not, however, work for graphs G' that contain cycles).

```
Function ComputeProbabilities(G')
    Data: G', a graph with components (V, E) and absorbing nodes
          D ⊆ V.
    foreach d ∈ D do
        ReverseSearch(G', d)
    end
Function ReverseSearch(G', d)
    Data: G', as above.
    Data: d, an absorbing node.
    Var excess[]         // array of numbers ∈ [0, 1] of size |V|
    Var stack
    excess[d.ID] = 1
    stack.push(d)
    while stack not empty do
        Var curNode = stack.pop()
        Var amt = excess[curNode.ID]       // probability to push
        foreach incoming edge curEdge of curNode do
            curEdge.map.IncrementOrAddProbability(d.ID, amt)
            excess[curEdge.src.ID] = amt * curEdge.probability
            stack.push(curEdge.src)
        end
        curNode.map.IncrementOrAddProbability(d.ID, amt)
        excess[curNode.ID] = 0
    end
```

Algorithm 1: Probability Calculation.

The *IncrementOrAddProbability*() function adds an amount of probability to the estimate of $P(d|c)$ stored in the map of component c, creating the map entry if it does not exist. The variable *excess* is a lookup table (indexed by node *ID*) that contains probability values for each node. (The variables *excess* and *amt* are designed to deal with overlapping paths).

In the worst case, the map at each node or link stores $|D|$ entries–one for each demand node in G'. Also in the worst case, every edge $e \in E$ in G has flow and is in G', yielding $O((|V| + |E|)|D|)$ in storage space. Since the *ReverseSearch* procedure is a modified DFS, the time required to compute probabilities for a given demand node is $O(|V|+|E|)$. Thus, calling *ReverseSearch* on all $|D|$ demand nodes in G' yields $O((|V| + |E|)|D|)$. Infrastructure networks typically have $|V| \approx |E|$ and $|D| \lessapprox \frac{1}{2}|V|$, giving **Algorithm 1** time and space complexity of $O(|V|^2)$.

The running time of the entire method is thus dominated by the flow generation step, which is typically more expensive than $O(|V|^2)$. The current paper uses the Edmonds-Karp algorithm (see [68]) for simplicity, which is $O(|V|^2|E|)$ on general graphs and $O(|V|^3)$ on infrastructure networks. Although flows can be generated using a variety of techniques (e.g., simulation), the method in **Algorithm 1** only applies if the transition graph G' is acyclic. Alternative methods (e.g., simulation, Markov chains) can be used if cycles are present.

3.4. An Algorithm for Interdependent Critical Flow Centrality

Given a model $\mathcal{S}$ with interdependent sub-systems $S_1, S_2, S_3, \ldots, S_n$, **Algorithm 1** can be used to compute CFC values for all components in each S_i at each time step t. This is not sufficient, however, as physical dependencies must be accounted for. Resource demands and criticality values must be propagated from one sub-system to the other.

Computation of the CFC for the entire model $\mathcal{S}$ proceeds by computing the CFC for each individual infrastructure system in topological order. Dependencies are processed from one system to the next in each iteration, passing demands from higher-level layers to lower-level ones. **Algorithm 2** provides a high level overview:

Function *ComputeInterdependentCFC($\mathcal{G}$)*

Data: $\mathcal{G}$, 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(G)
Var list ← TopologicalSort(V_G)
Var t ← 0
foreach S_i ∈ list do
    ComputeSingleSystemCFC(S_i)
end
```

Algorithm 2: Computing CFC for a set of interdependent infrastructures.

3.4.1. Converting Network Representations

As a pre-processing step, 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 [68, 69]).
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.

For step (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.

3.4.2. Computing CFC Values for a Sub-system

Computation of the CFC for sub-system S_i proceeds in two stages: (1) flow values and criticality values are propagated from other layers S_h ($h < i$) according to dependencies, and; (2) the CFC for S_i is computed using the technique discussed in **Section 3.3**. 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. Topological ordering ensures that S_h's criticality and flow values have been computed before S_i's. **Algorithm 3** provides an overview of single layer CFC computation.

Propagation of criticality and flow values proceeds by examining the set of relevant interconnection records:

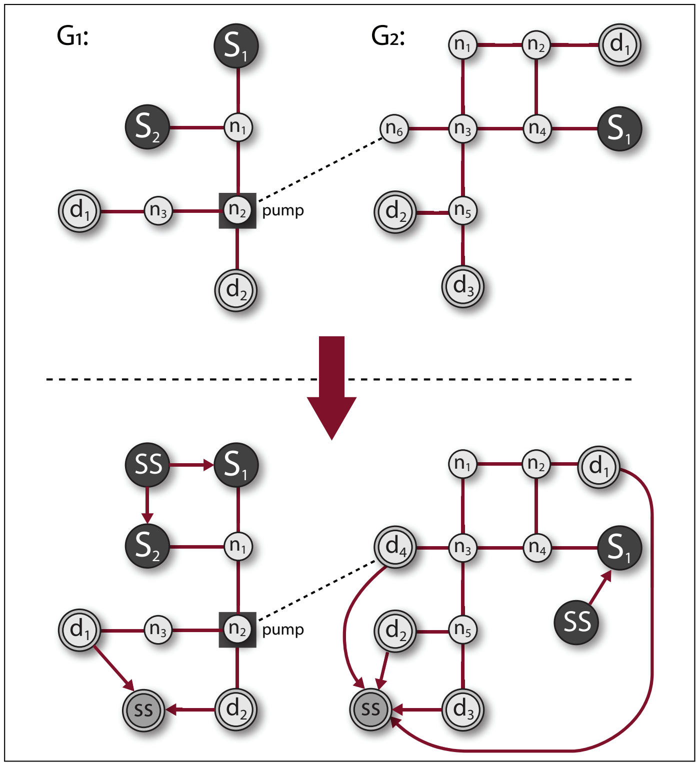

Fig. 3. Transformation of flow networks G_1 and G_2 with multiple sources (S_i) and sinks (d_i). Supersource ('SS') and supersink nodes ('ss') are added in the usual manner.

An interconnection record $IR(v_1, v_2)$ (where $v_1 \in S_h, v_2 \in S_i$) indicates a physical (resource) dependency between systems S_h and S_i (see **Figure 4**). 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 4 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 (e.g., the demand induced at v_2 is half of the flow at v_1).

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

Algorithm 3: Computing the CFC for a set of interdependent systems.

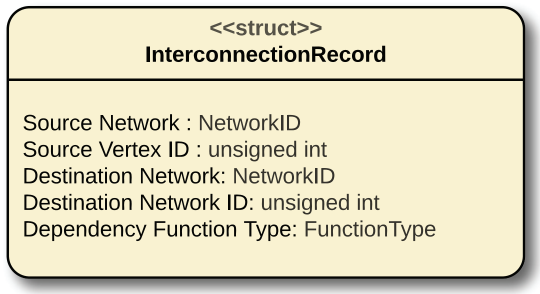

Fig. 4. An Interconnection Record used for Interdependencies.

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 4: Propagation of resource demands.

Figure 5 shows a 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 half of the flow—namely, 3031 and 2986 units.

Note also that edges and vertices with no flow are shown in black. The existence of such elements is an artifact of the Edmonds-Karp algorithm [69, 68] used in this simple instantiation, and one that would be corrected by using domain-specific methods (e.g., hydraulic simulation [58]).

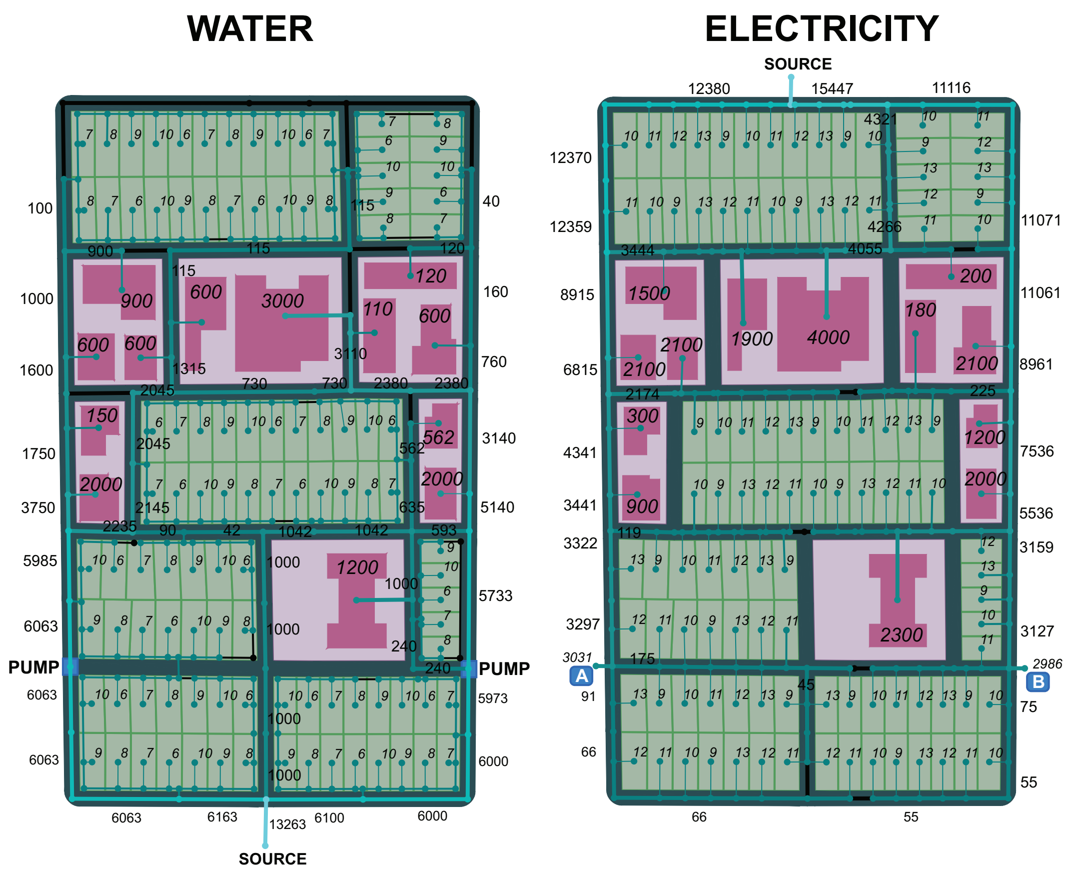

Fig. 5. Interdependent flows. Demand values are in italics while flow values are in regular font. Pumps in the water network are supplied with electricity by nodes A and B. Pumps require electricity proportional to half of their water flow. Black edges/vertices have zero flow.

Figure 6 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).

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. 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.

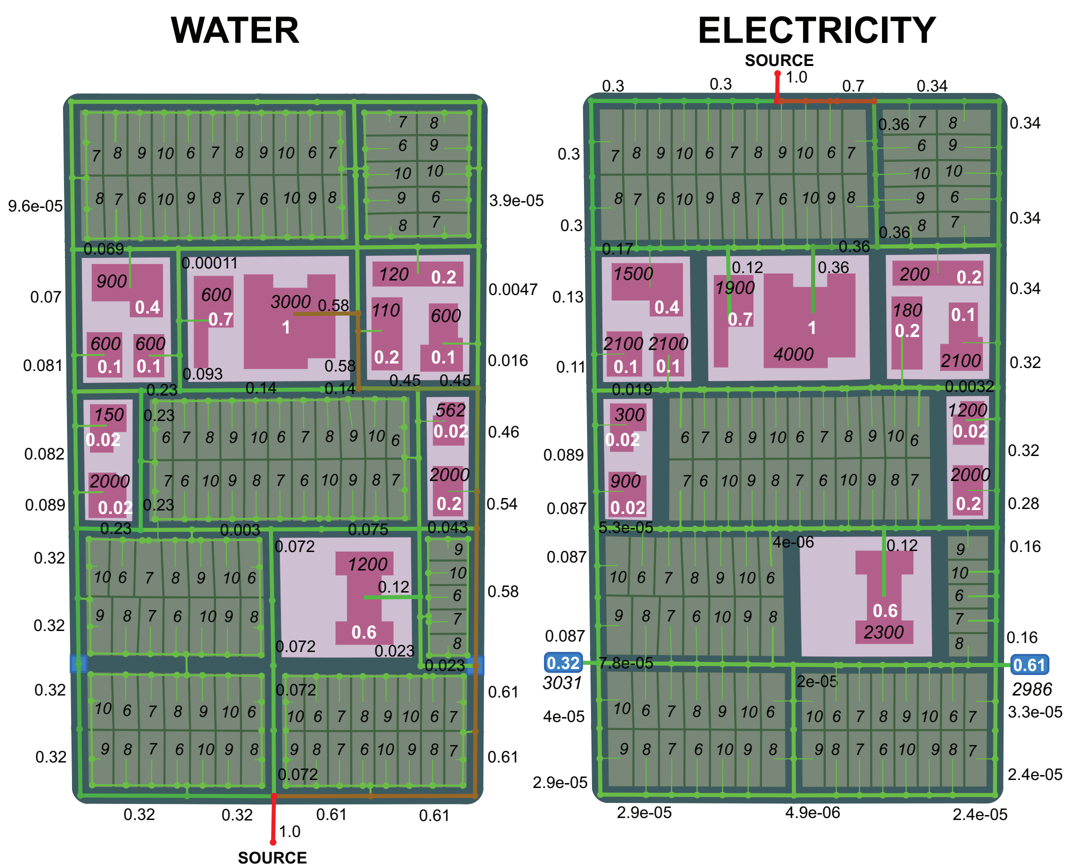

Fig. 6. (Normalized) critical flow centrality, computed from the flows in Figure 5. Demand values are in italics, CFC values are in regular font, and criticality ratings for buildings are in white. The electricity nodes that supply the water pumps are given criticality ratings of 0.32 and 0.61 and demands of 3031 and 2986 via Algorithm 4.

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 6** show that the electricity nodes supplying the pumps carry 16.7% and 8.7% of the total critical flow in the electricity network. It would be a poor decision to co-locate electrical assets with water assets when both are carrying highly critical flow.

4. Evaluation

This section demonstrates the method by means of a district-level model of a city containing electricity and water systems. The simplicity of the model is for explanatory purposes; it is possible to use the method on models of greater complexity, provided that the physical interdependencies create a directed, acyclic graph.

Each building/lot in the model is given: (1) a *type* (e.g., hotel); (2) a time series representing *hourly demand for water*; (3) a time series representing *hourly demand for electricity*, and; (4) a *criticality rating* in the interval $[0, 1]$. Time series are assumed to give average hourly demands over a 24-hour day. However, the method is general, and other scenarios could be supported, such as long-term (i.e., decadal) investigation of urban growth and its effect on capacity.

Time series data is assigned to buildings according to type (e.g., secondary school, restaurant), while lots are assigned time series randomly 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.

Empirical data for different types of buildings in summer was obtained from several sources (e.g., water consumption data was sourced from the California Public Utilities Commission [65], electricity data from Ontario Power Generation). Examples of water demand curves appear in **Figure 7** below:

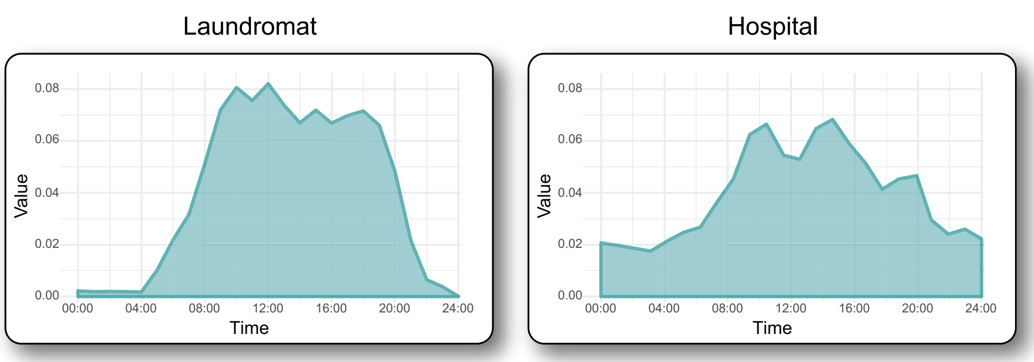

Fig. 7. Hourly time series showing water demands from a laundromat and hospital over an average day. The time series have been normalized to create a probability distribution. For use in the CFC method, these distributions are scaled by average water usage per day.

CFC values are computed for each time step $t \in [1, T]$ by loading the relevant time series data for t and executing **Algorithm 2**. An overview of the process is provided in **Algorithm 5**. 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.

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

Algorithm 5: Computing CFC on a system-of-systems with time-varying demands.

Figure 8 shows a graph of CFC values for the water network's edges over the full 24-hour cycle:

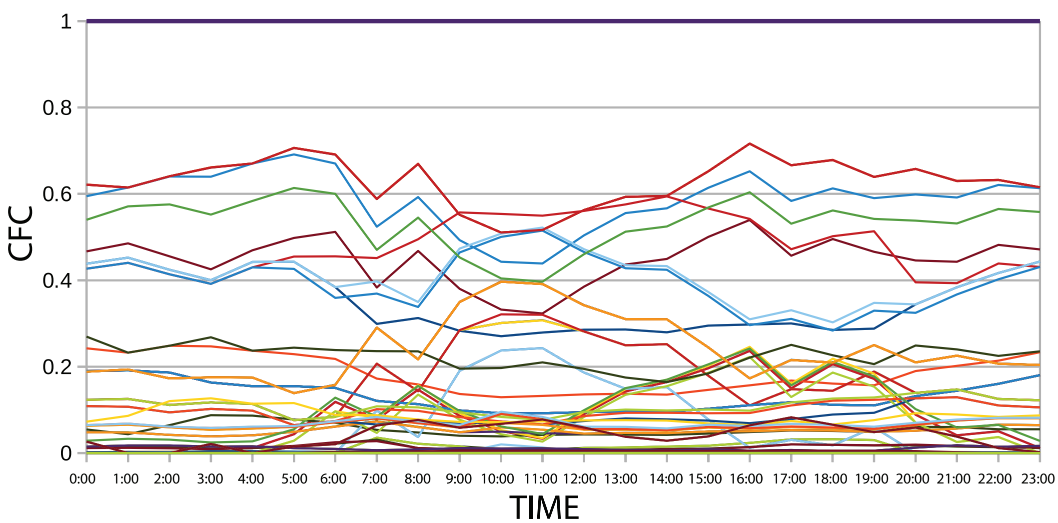

Fig. 8. CFC values for each edge in the water network.

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 9**, 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.

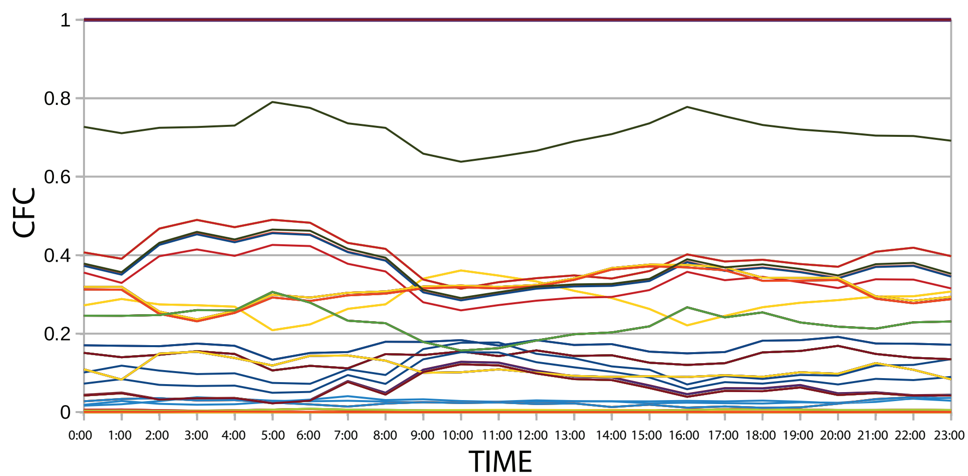

Fig. 9. CFC values for each edge in the electricity network.

To recap, Algorithm 5 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:

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

One major issue not addressed by classical works on network centrality (e.g., [9]) is the choice of ranking method for component measures taken at different times. The most intuitive approach to ranking the components is to take the *sample mean* of each column and to subsequently 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, information theory) 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 long-term behaviour of the system is of interest.

Figure 10 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.

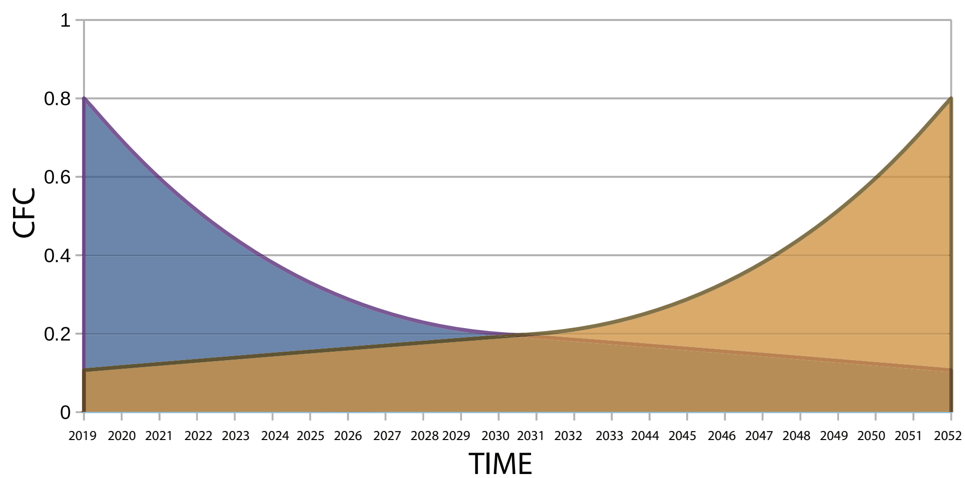

Fig. 10. Two components with similar integrals but different long term behavior.

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 11**) 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.

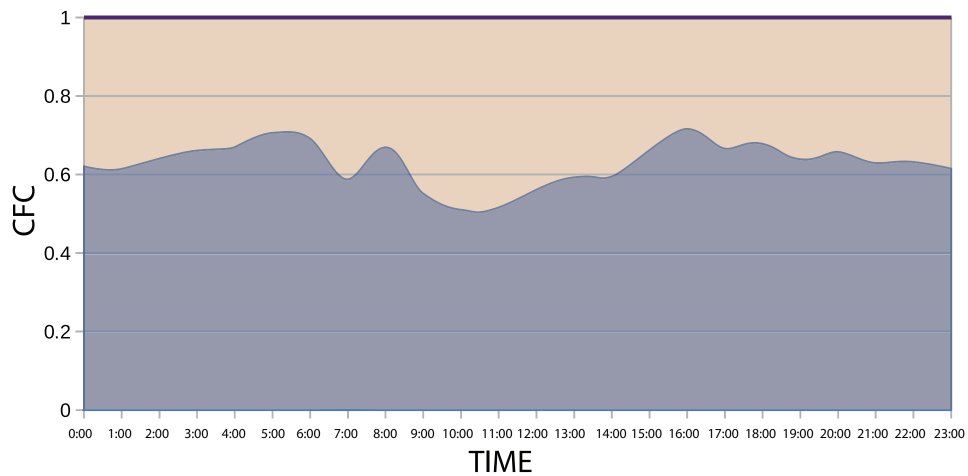

Fig. 11. Computing the global CFC value for a given component e. A cubic spline is overlaid on the CFC values for e. The integral of the spline (blue) is computed and normalized by the total area.

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

This section demonstrates that the CFC measure may be combined with standard approaches to network reliability—namely, (1) edge reliability measures, and; (2) "leave one out" failure analysis.

5.1. Edge Reliability

An arbitrary network model can be augmented by adding a **reliability function** $r : E \rightarrow [0, 1]$ that assigns edges $e \in E$ a **reliability rating** $r(e) \in [0, 1]$ [70]. One can combine this approach with CFC measures by creating a composite measure that estimates the joint reliability and criticality of a component. For instance, the (normalized) **Unreliable Critical Flow** ("UCF") for an edge $e \in V$ is:

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

where $C'^{CF}(e)$ is the normalized CFC for edge e. (The UCF is 'normalized' since values lie in the range $[0, 1]$, since $C'^{FC}(e)$ and $r(e)$ are both in $[0, 1]$.) Under this measure, components are important to the degree that they are: (1) unreliable, and; (2) instrumental for the delivery of resources to critical locations.

The computation of the UCF measure can be accomplished with a slight modification to the algorithm for the CFC. Instead of a static value $r(e)$, the reliability rating for a network component e can also be represented as a time series $R_e = \{r_{e1}, r_{e2}, \ldots, r_{ek}\}$. This allows the modeler to represent different processes (e.g., decreasing reliability of components over long time periods).

The UCF measures are computed for each timestep t using the CFC values and reliability ratings at t. 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 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. "Leave One Out" Failure Analysis

CFC measures can also be used with a common 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. A 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 (i.e., *fallbacks*) through which flow may travel in order to satisfy critical demand. **Figure 12** illustrates this situation.

This form of failure analysis provides an 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 network, one can assume that e is even more critical than the CFC measure alone might suggest.

Algorithm 6 shows a high-level view of a procedure in which capacities of edges in an 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

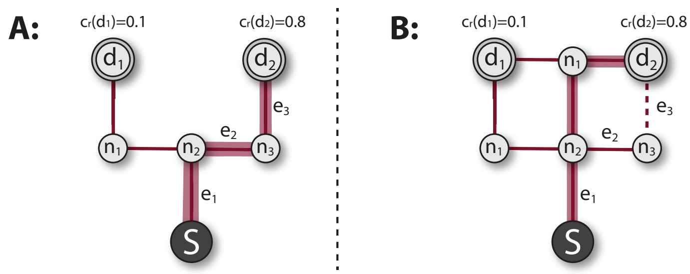

Fig. 12. Two networks with different behaviour in edge failure scenarios. Network A carries most of its critical flow through the path $\{e_1, e_2, e_3\}$. In case of edge failure, no alternative paths are available. Network B has a fallback route in case edge e_3 fails.

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 3.2.2) 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. Failure loss values range from 0 (no effect on resource delivery) to 1 (absolute disruption of resource delivery).

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

The edge failure mechanism was tested on the network from **Figure 5** 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 13 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.

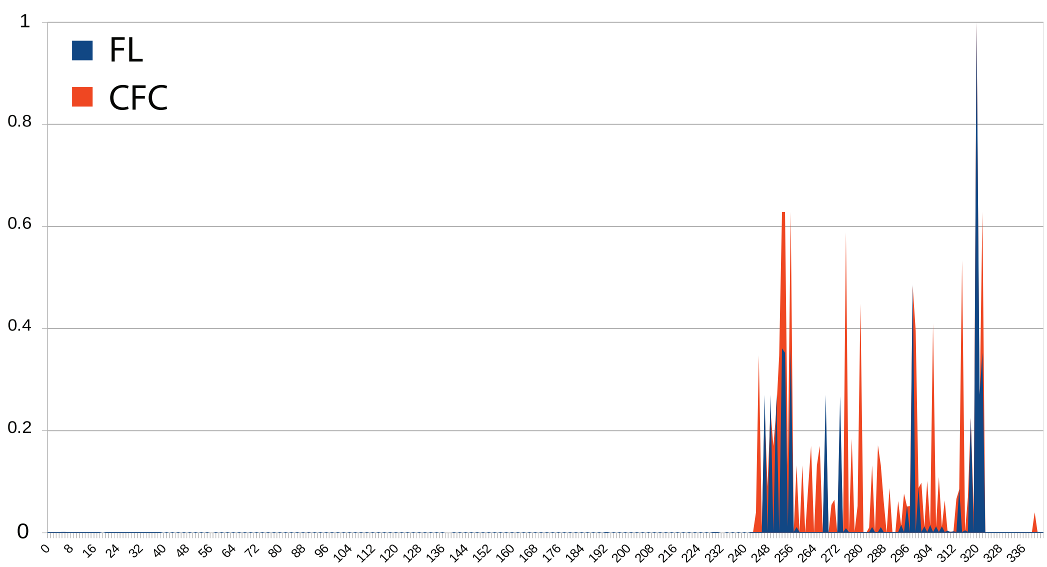

Fig. 13. Averaged *failure loss* (FL) and averaged *critical flow centrality* (CFC) on the edges of the water network in **Figure 5**, computed over a 24-hour period. The majority of edges (e.g., those that feed individual lots) have negligible FL and CFC values.

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 correlation 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. Reliability Integration Limitations and Assumptions

The edge failure analysis presented above was subject to several simplifying assumptions. First, geospatial dependencies were not included in the analysis for reasons of brevity. Second, the failure loss analysis is performed on each subsystem independently, given a flow solution for the entire system-of-systems; handling interdependencies requires iterative methods. Third, the reliability measures could also incorporate component capacity, in order to capture the intuition that a component nearing its maximum load is likely to be less reliable.

Fourth, 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 utilized in this paper, however, this assumption is reasonable.

This method outlined in this work does not assume that the methods used to model each layer are 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 a means of flow computation and a basic network topology. This design decision, while useful 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 demonstrated how component importance measures based on the notion of critical flow may be applied to interdependent, urban infrastructure systems. The motivation for the work was to provide urban planners and municipal engineers with a method of reasoning about the impacts of interventions on the flow of resources to critical locations. The main theme was that network analysis techniques could be combined with criticality and reliability metrics in order to produce composite methods that provide useful information to stakeholders.

The perspective of the method was resource-based, focusing on the ways in which system components participate in the delivery of resources. Each individual infrastructure system S_i of a composite system $\mathcal{S}$ was represented as a flow network with demands, capacities, supply limits, and criticality ratings. The paper considered physical dependencies in which one subsystem S_i requires resources from another subsystem S_j.

In the simple variant described in the paper, network flows and the *critical flow centrality* ("CFC") measure were computed using a discrete approach. More sophisticated variants are possible, including the use of domain-specific simulation techniques. For simplicity, the paper assumed that the subsystem dependencies form a directed acyclic graph.

The method was demonstrated by use of a simple, district-scale model of a city that contained electricity and water networks. Empirical data was used to estimate resource consumption for different types of buildings, yielding a set of demand curves that represent consumption 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.

Despite the simplifying assumption, the simple method presented in the paper satisfied the goals outlined in **Section 3.1**. First, the computation of CFC metrics for an interdependent system can be computed efficiently. For a model $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$ consisting of k subsystems, computation of CFC metrics for $\mathcal{S}$ on typical infrastructure networks is $O(V^2)$, where V is the average number of nodes in the subsystems. This compares favorably with other centrality measures, which can be $O(V^3)$ or greater.

Second, the demonstration showed that the basic method correctly propagates resource demand, criticality ratings and CFC values between systems. Not only are CFC values comparable across components within a given system, but they are commensurable across systems—even in cases where disparate modeling methodologies have been used.

Third, the paper showed how common network reliability approaches can be combined with CFC measures to yield composite metrics. Edge reliability can be directly integrated into the CFC framework by adding another attribute to the edges and tweaking the CFC computation slightly. The paper also discussed edge failure analysis, showing that a composite failure loss metric can be defined that gives an indication of the availability of fallback routes for the delivery of resources to critical locations.

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—that is, 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.

The version of the CFC computation presented in this paper is suitable for medium/long time horizons only, as it uses integer-valued representations for both 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 difficult to represent. As a consequence, rates of change (e.g., of flow) on system components cannot be analyzed precisely. The use of floating point representations and domain-specific flow computation methods (e.g., simulation) will avoid this limitation.

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 thank Eric Miller for the opportunity to work on this project.

References

[1] S. Dunn, S. M. Wilkinson, Identifying Critical Components in Infrastructure Networks Using Network Topology, Journal of Infrastructure Systems 19 (2012) 157–165.

[2] C. D. Nicholson, K. Barker, J. E. Ramirez-Marquez, Flow-based vulnerability measures for network component importance: Experimentation with preparedness planning, Reliability Engineering and System Safety 145 (2016) 62–73.

[3] Y. Almoghathawi, K. Barker, Component importance measures for interdependent infrastructure network resilience, Computers and Industrial Engineering 133 (2019) 153–164.

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

[5] K. Stephenson, M. Zelen, Rethinking centrality: Methods and examples, Social Networks 11 (1989) 1–37.

[6] V. Latora, V. Nicosia, G. Russo, Complex Networks Principles , Methods and Applications, Cambridge University Press, Cambridge (UK), 2017.

[7] M. Curado, L. Tortosa, J. F. Vicent, G. Yeghikyan, Analysis and comparison of centrality measures applied to urban networks with data, Journal of Computational Science 43 (2020) 101127.

[8] J. P. Scott, Social Network Analysis: A Handbook, SAGE Publications, London, UK, 2000.

[9] L. C. Freeman, S. P. Borgatti, D. R. White, Centrality in valued graphs: A measure of betweenness based on network flow, Social Networks 13 (1991) 141–154.

[10] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, S. Havlin, Catastrophic cascade of failures in interdependent networks, Nature 464 (2010) 1025–1028.

[11] A. Vespignani, Complex networks: The fragility of interdependency, Nature 464 (2010) 984–985.

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

[13] R. Zimmerman, C. E. Restrepo, The next step : quantifying infrastructure interdependencies to improve security, Int. J. Critical Infrastructures 2 (2006) 215–230.

[14] W. Kröger, C. Nan, Addressing Interdependencies of Complex Technical Networks, in: G. D'Agostino, A. Scala (Eds.), Networks of Networks: The Last Frontier of Complexity, Springer, 2014, pp. 279–310.

[15] R. M. Clark, S. Hakim, A. Ostfeld, Handbook of Water and Wastewater Systems Protection, Springer, 2011.

[16] J. Lopez, R. Setola, S. D. Wolthusen, Critical Infrastructure Protection: Information Infrastructure Models, Analysis, and Defense, Springer, 2012.

[17] A. Birolini, Reliability Engineering Theory and Practice, 5 ed., Springer, Berlin Heidelberg, 2007.

[18] N. R. Council, Drinking Water Distribution Systems: Assessing and Reducing Risks, Technical Report, Committee on Public Water Supply Distribution Systems: Assessing and Reducing Risks, 2006.

[19] D. Meijer, M. van Bijnen, J. Langeveld, H. Korving, J. Post, F. Clemens, Identifying critical elements in sewer networks using graph-theory, Water 10 (2018).

[20] A. A. Chowdhury, D. O. Koval, Power Distribution System Reliability: Practical Methods and Applications, Wiley, Hoboken, New Jersey, 2009.

[21] M. G. Resende, P. M. Pardalos, Handbook of Optimization in Telecommunications, Springer, New York, NY, 2006.

[22] M. A. P. Taylor, Vulnerability Analysis for Transportation Networks, Elsevier, Amsterdam, Netherlands, 2017.

[23] M. T. Thai, P. M. Pardalos, Handbook of Optimization in Complex Networks: Theory and Applications, volume 57, Springer, 2012.

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

[25] R. S. Wilkov, Analysis and Design of Reliable Computer Networks, IEEE Transactions on Communications 20 (1972) 660–678.

[26] I. Gertsbakh, Y. Shpungin, Network Reliability and Resilience, Springer, 2011.

[27] S. K. Chaturvedi, Network Reliability: Measures and Evaluation, Scrivener Publishing, Hoboken, New Jersey, 2016.

[28] B. Sudakov, V. Vu, Local resilience of graphs, Random Structures and Algorithms 33 (2008) 409–433.

[29] M. Krivelevich, C. Lee, B. Sudakov, Resilient pancyclicity of random and pseudo-random graphs, SIAM Journal on Discrete Mathematics 24 (2009) 1–17.

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

[31] M. Papa, Critical Infrastructure Protection II, Springer, 2008.

[32] A. Garas, Interconnected networks, Springer, 2016.

[33] F. Petit, D. Verner, J. Phillips, L. P. Lewis, Critical Infrastructure Protection and Resilience—Integrating Interdependencies, in: A. J. Masys (Ed.), Security by Design, Springer, 2018, pp. 193–219.

[34] M. Ouyang, Review on modeling and simulation of interdependent critical infrastructure systems, Reliability Engineering and System Safety 121 (2014) 43–60.

[35] G. Bianconi, Multilayer Networks: Structure and Function, Oxford University Press, 2019.

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

[37] Per Hokstad, I. B. Utne, J. Vatn, Risk and Interdependencies in Critical Infrastructures: A Guideline for Analysis, Springer, 2012.

[38] G. D'Agostino, A. Scala, Networks of Networks: The Last Frontier of Complexity, Springer, 2014.

[39] H. Amini, K. G. Boroojeni, S. S. Iyengar, P. M. Pardalos, F. Blaabjerg, A. M. Madni, Sustainable Interdependent Networks: From Theory to Application, Springer, 2018.

[40] M. H. Amini, K. G. Boroojeni, S. S. Iyengar, P. M. Pardalos, F. Blaabjerg, A. M. Madni, Sustainable Interdependent Networks II: From Smart Power Grids to Intelligent Transportation Networks, Springer, 2019.

[41] S. M. Rinaldi, J. P. Peerenboom, T. K. Kelly, Identifying, understanding, and analyzing critical infrastructure interdependencies, IEEE Control Systems Magazine 21 (2001) 11–25.

[42] A. Nieuwenhuijs, E. Luiijf, M. Klaver, Modeling Dependencies in Critical Infrastructures, in: M. Papa, S. Shenoi (Eds.), Critical Infrastructure Protection II, Springer, 2008, pp. 205–213.

[43] E. E. Lee, J. E. Mitchell, W. 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 (2007) 1303–1317.

[44] J. P. Peerenboom, R. E. Fisher, Analyzing cross-sector interdependencies, in: Proceedings of the Annual Hawaii International Conference on System Sciences, 2007, pp. 1–9.

[45] G. E. Apostolakis, D. M. Lemon, A screening methodology for the identification and ranking of infrastructure vulnerabilities due to terrorism, Risk Analysis 25 (2005) 361–376.

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

[47] V. Latora, M. Marchiori, Efficient Behavior of Small-World Networks, Physical Review Letters 87 (2001) 3–6.

[48] M. Newman, Networks: An Introduction, 2 ed., Oxford University Press, 2018.

[49] G. Galvan, J. Agarwal, Vulnerability analysis of interdependent infrastructure systems, in: T. Haukaas (Ed.), 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.

[50] N. K. Svendsen, S. D. Wolthusen, Connectivity models of interdependency in mixed-type critical infrastructure networks, Information Security Technical Report 12 (2007) 44–55.

[51] N. K. Svendsen, S. D. Wolthusen, Analysis and statistical properties of critical infrastructure interdependency multiflow models, Proceedings of the 2007 IEEE Workshop on Information Assurance, IAW (2007) 247–254.

[52] N. K. Svendsen, S. D. Wolthusen, Graph models of critical infrastructure interdependencies, Lecture Notes in Computer Science 4543 LNCS (2007) 208–211.

[53] N. K. Svendsen, S. D. Wolthusen, An analysis of cyclical interdependencies in critical infrastructures, Lecture Notes in Computer Science 5141 LNCS (2008) 25–36.

[54] P. Kotzanikolaou, M. Theoharidou, D. Gritzalis, Interdependencies between critical infrastructures: Analyzing the risk of cascading effects, in: S. Bologna, B. Hämmerli, D. Gritzalis, S. Wolthusen (Eds.), Critical Information Infrastructure Security, Springer, 2013, pp. 104–115.

[55] G. Stergiopoulos, P. Kotzanikolaou, M. Theocharidou, D. Gritzalis, Risk mitigation strategies for critical infrastructures based on graph centrality analysis, International Journal of Critical Infrastructure Protection 10 (2015) 34–44.

[56] M. Shahraeini, P. Kotzanikolaou, A dependency analysis model for resilient wide area measurement systems in smart grid, IEEE Journal on Selected Areas in Communications 38 (2020) 156–168.

[57] J. Williams, Identifying Sensitive Components in Infrastructure Networks via Critical Flows, engrXiv (2019). URL: engrxiv.org/hyzbx.

[58] P. Novak, V. Guinot, A. Jeffrey, D. E. Reeve, Hydraulic Modelling – an Introduction: Principles, Methods and Applications, Spon Press, 2010.

[59] P. Stride, Super sewer: an introduction to the Thames Tideway tunnel project in London, Proceedings of the Institution of Civil Engineers - Civil Engineering 169 (2016) 51–51.

[60] P. Stride, The Thames Tideway Tunnel: Preventing Another Great Stink, The History Press, 2019.

[61] M. Amin, Toward Secure and Resilient Interdependent Infrastructures, Journal of Infrastructure Systems 8 (2007) 67–75.

[62] R. F. Austin, D. P. DiSera, T. J. Brooks, GIS for critical infrastructure protection, CRC Press, 2016.

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

[64] M. Mair, J. Zischg, W. Rauch, R. Sitzenfrei, Where to find water pipes and sewers? - on the correlation of infrastructure networks in the urban environment, Water 9 (2017).

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

[66] R. H. Shumway, D. S. Stoffer, Time Series Analysis and Its Applications, Springer, 2017.

[67] R. Setola, S. Bologna, E. Casalicchio, V. Masucci, An Integrated Approach for Simulating Interdependencies, in: M. Papa, S. Shenoi (Eds.), Critical Infrastructure Protection II, Springer, 2008, pp. 229–242.

[68] T. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 3 ed., MIT Press, Cambridge, MA, 2009.

[69] R. Ahuja, T. Magnanti, J. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, Upper Saddle River, NJ, 1993.

[70] E. Zio, From complexity science to reliability efficiency: a new way of looking at complex network systems and critical infrastructures, International Journal of Critical Infrastructures 3 (2007) 488.