Identifying Sensitive Components in Infrastructure Networks via Critical Flows

James Williams, J.D.[1]

[1]Department of Computer Science, University of Toronto, Toronto, Canada. Email: jamesbw@cs.toronto.edu

ABSTRACT

This paper introduces a novel set of component importance measures that are based on the concept of critical flow. Various research communities have developed techniques for identifying critical components of networks. The methods in this paper extend previous work on flow-based centrality measures by adapting them to the assessment of critical infrastructure in urban systems. The motivation is to provide municipalities with a means of reasoning about the impact of urban interventions. An infrastructure system is represented as a flow network in which demand nodes are assigned both demand values and criticality ratings. Sensitive elements in the network are those that carry critical flows, where a flow is deemed critical to the extent that it satisfies critical demand. A method for computing these flows is presented, and its utility is demonstrated by comparing the new measures to existing flow centrality measures. The paper also shows how the method may be combined with standard approaches to reliability analysis.

INTRODUCTION

This paper describes a set of *component importance measures* ("CIMs") that allow system components to be ranked according to their role in the delivery of resources to critical locations. These measures can be viewed as variants of *flow centrality* (Freeman 1977) that are customized for flow networks in which demand nodes are assigned both demand values and criticality ratings. The paper describes a general method for computing critical flow CIMs, as well as a simple embodiment that demonstrates the various stages of computation.

The perspective on infrastructure criticality adopted in this work is *demand-based*, motivated by an interest in *decision support systems* (“DSS") (Holsapple 2008) for use by municipal governments [see (Geertman et al. 2015; Brail and Klosterman 2001)]. As urbanization continues at a rapid pace (UN-Habitat 2009; Smil 2019) cities are experiencing significant increases in population and concomitant increases in demand for goods and services. The resulting capacity issues are a major source of risk for legacy infrastructure systems due to the fact that they are typically extremely expensive to alter [e.g., the Thames Tideway tunnel project (Stride 2016; Stride 2019)].

In addition to population growth, various interventions have the potential to alter patterns of demand, and therefore to change the distribution of flows within infrastructure systems. For example, densification involves constructing new residential apartment units on top of legacy infrastructure systems (Kishewitsch 2015), creating new demands that alter existing flow patterns and potentially change the set of components that are relied upon to deliver resources to existing critical locations. As a second example, maintenance activities can temporarily impact the availability or capacity of existing network components, leading to similar consequences.

In the worst case, these factors may impose undue stress on existing infrastructure. Increases in demand (or reductions in capacity) may leave critical locations such as hospitals with reduced access to resources. In addition, resource flow to critical locations may be routed through less reliable infrastructure components. In some cases, there may be no fallback routes available in case of component failure.

This paper provides a method for identifying sensitive portions of an infrastructure system according to the degree to which those portions carry flow to critical locations (i.e., **critical flows**). The critical infrastructure protection literature contains methods for identifying vulnerabilities in containing critical assets [e.g., (Apostolakis and Lemon 2005)]. The method presented in this paper allows criticality ratings to propagate through the entire network. It is also highly granular, intended for use in urban decision support systems operating at a parcel level. By combining critical flow with standard reliability techniques, stakeholders can visualize the portions of an infrastructure system that: (1) play a disproportionate role in delivering resources to critical locations, and; (2) lack backup/failsafe routes.

Of course, infrastructures do not exist in isolation. Interdependencies between infrastructure systems are ubiquitous, leading to complex failure patterns that are much more difficult to predict (Amin 2007). The resulting 'systems-of-systems' (DeLaurentis and Crossley 2005) are subject to additional risks that arise from the complex interaction of their component systems, including *cascading failures* (Rinaldi et al. 2001). Nevertheless, in the interests of brevity this paper focuses on a single infrastructure system at a single moment in time.

The structure of this paper is as follows. Section Two provides a concise review of background information on network component importance measures, as well as a discussion of the most relevant applications to infrastructure systems. Section Three discusses the rationale for introducing flow criticality. Section Four provides a simple instantiation of the method described above, while Section Five presents a comparison with other centrality measures and a demonstration of how reliability analysis might be integrated with critical flow modeling. The paper closes with numerous suggestions for future work.

BACKGROUND AND PREVIOUS WORK

Many research communities have proposed *component importance measures* ("CIMs") (Almoghathawi and Barker 2019) for infrastructure systems. Examples of CIMs include network efficiency (Nagurney and Qiang 2009), flow vulnerability (Ouyang et al. 2014), and flow capacity rate (Nicholson et al. 2016). Of particular interest are the *centrality measures*, which are used to identify the most central components in a network (Stephenson and Zelen 1989; Borgatti 2005; Borgatti and Everett 2006; Latora et al. 2017; Newman 2018). Arranged in categories [e.g,. (Scott 2000; Ouyang et al. 2014)], these include:

1. *Nearness measures*, which calculate a component's centrality by means of its proximity to other components. Examples include: degree centrality (Newman 2018), closeness

centrality (Borgatti 1995), residual closeness centrality (Dangalchev 2006), information centrality (Latora and Marchiori 2007), and evidential centrality (Wei et al. 2013).

2. *Betweenness measures*, which consider components to be central to the extent to which they stand between other components as intermediaries. Examples include: (shortest path) betweenness centrality (Freeman 1977), (general path) betweenness centrality (Izquierdo and Hanneman 2006), load centrality (Goh et al. 2001), and random walk betweenness centrality (Newman 2005).
3. *Dynamical measures*, which examine both topology and dynamical processes situated on the network. Examples include: flow centrality (Freeman et al. 1991), traffic load centrality (Newman 2001), random walk centrality (Noh and Rieger 2004; Newman 2005), routing betweenness centrality (Dolev et al. 2010), dynamical influence (Klemm et al. 2012), efficiency centrality (Wang et al. 2017), and percolation centrality (Piraveenan et al. 2013).

Flow-based Centrality Measures

Several centrality measures are based on the concept of network flow. The progenitor of this family is the classical notion of *flow centrality* (Freeman et al. 1991). 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. 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:

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

This can be normalized by dividing the flow that passes through v by the total flow between all other pairs of nodes, yielding the percentage of flow that depends on v:

$$C'^F(v) = \frac{C^F(v)}{\sum_{u \neq w \neq v} m_{u,w}} \tag{2}$$

Since the analysis of supply/demand networks focuses primarily on the relation between source and sink nodes, this paper introduces a narrower formulation of flow centrality [and of *edge-flow centrality* (Nicholson et al. 2016)]. The **source-sink flow centrality** (“SSFC") of a node v is the degree to which the maximum flow between all source and demand nodes depends on v:

$$C^{SSF}(v) = \sum_{s \in S, d \in D, s \neq d \neq v} m_{s,d}(v) \tag{3}$$

where $m_{s,d}$ is the maximum flow between s and d, and $m_{s,d}(v)$ is the maximum flow between s and d that depends on v. This can be normalized by dividing the flow that passes through v by the total flow between all pairs of source/sink nodes:

$$C'^{SSF}(v) = \frac{C^F(v)}{\sum_{s \in S, d \in D} m_{s,d}} \tag{4}$$

Flow-based CIMs Applied to Infrastructure

Various research communities have applied centrality measures to the study of infrastructure systems [see, for example, (Murray et al. 2003; Zio 2007; Cadini et al. 2009; Lü et al. 2016)]. Recent work has focused on dynamical CIMs, including those based on network flows. Comparisons between topological and flow-based methodologies have appeared in several recent studies [e.g., (Zio and Kroger 2009; Ouyang 2013; Ouyang et al. 2014)].

Probabilistic estimates of reliability in a flow network were provided by (Kansal and Kumar 1995). A network is deemed *reliable* to the extent that demand nodes have at least one path to the source, given component failures. To determine reliability, an *appended spanning tree* data structure is used to identify a set of disjoint spanning trees. For a given failure scenario, each of these spanning trees T_i is inspected to determine if demand nodes have at least one path to the source. The results are then aggregated to create a measure of reliability for the scenario. The network is not a proper flow network with capacities, but the concept of fallback routes is evident.

A method for ranking vulnerabilities in a directed, capacitated, flow network is provided in (Michaud and Apostolakis 2006). Damage scenarios are created by disabling one of the edges. A scenario is evaluated by determining its impact on the *supply coverage* (supply/demand ratio) at each sink. A greedy, path-based algorithm finds the geodesic with maximum capacity between a sink and a source, and reserves the demand from the sink against the capacity of each edge and vertex of the geodesic. (If capacity constraints are violated, there is a supply problem for the given sink). Multi-attribute decision theory is used to calculate the *disutility* of the scenario. There is no priority order imposed on the nodes, nor is there consideration of fallback routes.

A CIM for a directed, uncapacitated, flow network is provided in (Dunn and Wilkinson 2012). To identify critical nodes: (1) a synthetic baseline network G_B is generated, using nodal degree to identify a unique supply node and to assign demand values; (2) hydraulic simulation is used to compute flows; (3) a 'roving supply node' simulation is performed. In each iteration a single demand node d is removed from G_B, yielding $G_B - \{d\}$, and hydraulic simulation is used again. After iterating through all demand nodes, a new supply node is chosen and the process is repeated until all combinations have been covered. The impact of each scenario j is $\sqrt{(\Delta_{1j})^2 + (\Delta_{2j})^2 + \cdots + (\Delta_{nj})^2}$, where Δ_{ij} is the difference in flow for node i between G_B and $G_B - \{i\}$. Unlike the present paper, the supply nodes are not fixed, and no priority is imposed on demand nodes.

A hierarchical approach to evaluating the robustness of capacitated flow networks is provided in (Ferrario et al. 2016). A network is represented as a multi-level flow model in which each demand node is assigned a *priority level*. Reliability evaluation is carried out through a Monte Carlo approach (Zio 2013) in which a system configuration is created by sampling the link capacities from their probability distributions. For each configuration, a greedy, surplus-based algorithm pushes flow through the network, satisfying higher priority nodes first. The results from numerous such configurations are used to estimate the probability distribution of the amount of flow delivered to each demand node at equilibrium. The network is *robust* to the extent that sufficient levels of flow reach the demand nodes despite capacity degradation of the links. The decision to assign flow based on node priority is a major difference from the method described in this paper.

Flow-based vulnerability measures for a capacitated and directed network are developed in (Nicholson et al. 2016). The measures are used to identify a set of list of network edges that can be hardened against damage from geographically-situated disruptions; Performance of the network is measured by computing an *all-pairs average network flow* $\varphi(x)$ by means of a MINIMUM COST NETWORK FLOW ALGORITHM (Ahuja et al. 1993). Several edge importance measures are considered, including: (1) *all-pairs max-flow edge count*, the total number of times a given edge is utilized in all $s - t$ pairs max-flow problems, and; (2) *edge flow centrality*: sum of flow on the edge for all possible $s - t$ pair max-flow problems, normalized by the sum of all-pairs max-flows. No priority order is imposed on nodes, but the measures are important precursors for the work described in this paper.

Finally, a variant of *current-flow betweenness centrality* (Newman 2005) was applied to to urban networks in (Agryzkov et al. 2019). Since the computational cost of this type of centrality measure can be quite high, an approximation method is used to allow for its use on high density urban datasets. The authors use the new method to study pedestrian flow on a dataset of 186 nodes that is drawn from a real-world street network in Spain. Priority order on nodes is not supported, but the method is a useful comparator as it is designed for analyzing urban infrastructure at district scale.

While this is not an exhaustive list of previous work on flow-based CIMs, it is representative of basic techniques and guiding assumptions. Although various authors have proposed means of identifying critical components using flow-based techniques, only one (Ferrario et al. 2016) permits network components to be labeled with criticality values of the sort commonly used in critical infrastructure protection [e.g., (Apostolakis and Lemon 2005)]. In that work, the algorithm for generating the flow solution ensures that higher priority nodes are satisfied first. In contrast, the current work does not make any such assumption, allowing a variety of techniques (e.g., hydraulic simulation) to be used in determining flow solutions and the resulting centrality values.

CRITICAL FLOWS

At a high level, the method in this paper provides a means of modeling the impact of changes in demand, supply, or capacity on the distribution of flows within a system. Of particular interest are those system components that deliver resources to critical locations such as hospitals. Urban planners and other municipal stakeholders could benefit from software artifacts that permit the visualization of planning interventions on the flow of resources within a system.

For example, changes in demand can result in significant changes to the distribution of flows, including **critical flows** (i.e., flows that supply critical locations with resources). Part A of **Figure 1** shows a capacitated network in which demand node d_1 has been deemed to be critical. Part B shows a maximum flow solution for the network; the flow to d_1 is balanced between the paths (S, n_1, d_1) and (S, n_2, d_1), with each path carrying 7.5 units of flow. Furthermore, the flow to all demand nodes is approximately balanced, and the internal edge (n_1, n_2) is not used.

A slight change to the network drastically alters the distribution of critical flows. Part C of **Figure 1** shows a modified network in which the demand at d_3 has been increased to 22. Part D shows the resulting change in critical flow patterns: the majority of flow to critical location d_1 is now routed through the upper path (S, n_1, d_1). If, for instance, those network elements were unreliable or scheduled for maintenance, the introduction of new demand at d_3 might increase the risk that critical locations could experience resource shortages.

Representing Criticality

There are many possibilities for defining CIMs based on critical flows. One major design decision is the representation of criticality, for which there are three salient options: (1) a *binary representation*, in which a demand node is either critical or not; (2) a *categorical representation*, in which a demand node's criticality is selected from a finite set of categories (e.g., $\{low, med, high\}$), and; (3) a *continuous representation*, in which a demand node's criticality is in $\mathbb{R}$.

Given a representation of criticality, one must provide a measure of importance. There are several possible ways to measure a component's importance: (1) *absolute critical flow*, the amount of critical flow that passes through it; (2) *percentage critical flow*, the percentage of its flow that is critical; (3) *location count*, the number of critical locations that receive flow through paths in which that component appears, and; (4) *weighted expected critical flow*, the expected amount of resource that it supplies to demand nodes, weighted by their criticality.

COMPONENT IMPORTANCE MEASURES BASED ON CRITICAL FLOWS

The concept of critical flow may be used to define numerous CIMs. Two such measures are described in the following sections.

Critical Flow Centrality

The most simple means of applying the concept of critical flow to CIMs is to formulate a new measure of *flow centrality* (Freeman 1977). The underlying idea is that a node is deemed central to the extent that it is involved in the flow of resources to critical locations. Examples of such measures include:

1. **(Binary) Critical Flow Centrality**: let $D_C \subseteq V = \{d_1, d_2, \ldots, d_k\}$ be the set of demand nodes in G that are deemed critical. and let $S = \{s_1, s_2, \ldots, s_p\}$ be the set of source nodes in G. Let $m_{s,d}$ be the maximum flow between $s \in S$ and $d \in D_C$. (In the case where there is a single source, this is the same as the flow entering d). Furthermore, given node $v \in V - D_c$, let $m_{s,d}(v)$ be the maximum flow between s and d that depends on v. Then the **(binary) critical flow centrality** of a node $v \in V - D_c$ is the degree to which the maximum flow between critical demand nodes and source nodes depends on v:

$$C^{CF}(v) = \sum_{s \in S, d \in D_C} m_{s,d}(v) \tag{5}$$

2. **Weighted Critical Flow Centrality**: let the set of demand nodes in the network be $D = \{d_1, d_2, \ldots, d_m\}$, and consider function $c_r : D \to \mathbb{R}^+$ that maps a demand node $d \in D$ to a *criticality value* $c_r(d)$. Then the **weighted critical flow centrality** of a node $v \in V$ is:

$$C^{WCF}(v) = \sum_{s \in S, d \in D} c_r(d) m_{s,d}(v) \tag{6}$$

Both of these CIMs may be normalized by the total critical flow on the network to yield measures in the interval $[0, 1]$.

Probabilistic Critical Flow Measures

The measures introduced above are based on maximum flows. One can relax this assumption, creating another type of CIM in which demand nodes have both demand values and criticality levels. Consider demand nodes $D = \{d_1, d_2, \ldots, d_m\}$, and criticality function $c_r : D \to \mathbb{R}^+$. Let $\mathcal{D}$ be a family of functions $\delta_i : D \to \mathbb{R}^+$, each of which assigns a **demand** $\delta(d) \in \mathbb{R}^+$ for each demand node $d \in D$. Finally, let an **assignment** A be a concrete choice of function δ_i from $\mathcal{D}$, and let $f_A(d)$ be the **flow received** by $d \in D$ under A.

The **flow** in network G given assignment A is the aggregate of all flows reaching the demand nodes:

$$F_A(G) = \sum_{d \in D} f_A(d) \tag{7}$$

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

$$F_A^C(G) = \sum_{d \in D} f_A(d) c_r(d) \tag{8}$$

Let $f_A(v_i, d_j)$ be the flow that reaches $d_j \in D$ from/through node $v_i \in V$ under assignment A. The main quantity of interest is the expected amount of flow from v_i that reaches d_j:

$$E[f_A(v_i, d_j)] \tag{9}$$

Assuming that there is a method to compute these expectations, one can define a relevant CIM. For instance, the **(probabilistic) critical flow centrality** (“CFC") of a vertex $v \in V$ is:

$$C^{PCF}(v) = \sum_{d \in D} c_r(d) E[f_A(v, d)] \tag{10}$$

This is the sum of all expected amounts of flow being delivered via v to the demand nodes, weighted by their criticality. This quantity may be normalized by an appropriate expression, such as the total flow $F_A(G)$ or the total flow weighted by criticality:

$$C'^{PCF}(v) = \frac{C^{PCF}(v)}{F_A^C(G)} = \frac{\sum_{d \in D} c_r(d) E[f_A(v, d)]}{\sum_{d \in D} c_r(d) f_A(d)} \tag{11}$$

Variations on these formulas are easy to construct. First, identical definitions can be formulated for edges. Second, one can concoct unweighted versions in order to accommodate the boolean representation of criticality. Third, additional components may be incorporated (e.g., reliability estimates for components, as discussed in Section 6). Lastly, these definitions are compatible with the use of *utility functions*, although that topic is not discussed in this paper.

COMPUTING THE CRITICAL FLOW MEASURES

The previous section provided definitions of new CIMs, but no guidance on how they are to be computed. This section presents a general framework for doing so, as well as a simple instantiation that computes the (probabilistic) *critical flow centrality* (“CFC").

A Method for Critical Flow Analysis

The framework for computing critical flow measures requires that the following four elements are provided:

1. a representation of an infrastructure system as a flow network;
2. criticality metadata for nodes (i.e., demand values, criticality ratings, capacity constraints);
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;

Since these elements can be supplied in different ways, there are many possible instantiations ranging from simple to highly complex. This paper presents a simple instantiation (a.k.a., the *present* instantiation) for reasons of brevity.

Network Representation

In the present instantiation, an infrastructure system is represented as a weighted, capacitated, flow network $G = \langle V, E \rangle$ where V is a set of nodes, and $E \subseteq V \times V$ is a set of links. Each node $v \in V$ is situated in Euclidean space, having **world coordinate** $\vec{w}(v) = (v_x, v_y, v_z) \in \mathbb{R}^3$. Each link $e = (v_i, v_j) \in E$ 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 $\|\vec{w}(v_i) - \vec{w}(v_j)\|_2$. Bi-directional relationships, cycles, and self-loops are all permitted.

The network G contains both source (supply) and sink (demand) nodes. The set of **source nodes** is $S = \{s_1, s_2, \ldots, s_p\} \subseteq V$, and the set of **demand nodes** is $D = \{d_1, d_2, \ldots d_m\} \in V$. All other nodes are called *transmission nodes*. A **flow** on G is a real-valued function $f : E \rightarrow \mathbb{R}$ on G's links 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 nodes $v_t \in V - (D \cup S)$, we have $\sum_{v \in V} f((v_t, v)) = 0$.

The **value of a flow** is defined as the flow exiting the source nodes: $|f| = \sum_{v \in V} f(s, v)$. While a network with multiple source and sink nodes may be reduced to a network with a single sink and source, [see (Cormen et al. 2009)], the explicit representation is used throughout this paper.

Demand Distributions and Criticality

The basic network flow model is augmented with additional metadata. First, a **demand function** $\delta : D \rightarrow \mathbb{N}$ maps a demand node $d \in D$ to a demand $\delta(d)$. Second, a **supply constraint function** $f_s : S \rightarrow \mathbb{N}$ assigns each source node $s \in S$ a maximum flow $c_f(s)$. Third, a **criticality function** $cr : D \rightarrow [0, 1]$ maps $d \in D$ to a criticality rating $cr(d)$ between 0 and 1.

An **assignment** to a network involves assigning supply constraints to all source nodes and criticality/demand values to all demand nodes. Theses values may be provided by probability distributions (e.g., those arising from stochastic processes), or via external data (e.g., real world data obtained from smart meters).

Calculating Network Flow

For a given assignment to the network, a network flow must be computed. While binary and categorical representations of criticality can be accommodated using a *multi-commodity network flow* algorithm (Ahuja et al. 1993), the continuous representation of criticality requires the use of alternative techniques, including: (1) general network flow algorithms, or (2) *domain-specific simulation techniques* [e.g., hydraulic simulation (Novak et al. 2010)].

The most intuitive way to create a flow on an network is to utilize an algorithm for the MAXIMUM

FLOW PROBLEM (Ahuja et al. 1993). A major drawback to this approach is that popular maxflow algorithms do not yield network flows that are *balanced* across competing paths. [Algorithms that allow for disjunctive constraints exist, as in (Pferschy and Schauer 2013), but they have some drawbacks]. Domain-specific methods such as hydraulic simulation (Novak et al. 2010) are preferable in cases where a realistic flow must be obtained. For simplicity, this paper uses the the Edmonds-Karp algorithm [see (Cormen et al. 2009)].

Calculating Critical Flow from Flow Patterns

Once a flow solution has been obtained for the current assignment A, the CIM can be computed. In the case of the CFC measure, one must compute $E[f_A(c_i, d_j)]$ (Equation 9), the expected amount of flow passing through component c_i that ends up at demand node d_j. For a model that uses empirical data, a sampling approach is required. The demonstration method in this paper represents an infrastructure system at a single point in time using integer-valued demands, so that Equation 9 is equivalent to the probability that a unit of flow passing through c_i ends up at d_j, multiplied by the total amount of flow passing through c_i:

$$E[f_A(c_i, d_j)] = P(d_j|c_i) f_A(c_i)$$

The required probabilities can be computed in several ways. The most intuitive approach is to reduce the problem to *Markov chain* computation (Stewart 1994; Stewart 2009). Another feasible solution is to estimate probabilities from the flow network using a *Time Forward Random Walk* (Blanchard and Volchenkov 2011). In contrast, the present instantiation uses a graph search approach to directly compute the probability for each node v (or link e) that a unit of resource passing through v (or e) lands in a given demand node d. **Figure 2** shows how this is performed for a demand node corresponding to a lot/parcel.

The flow solution is transformed into a secondary **transition graph** G' that represents the flow in terms of transition probabilities. Every node in G' contains a list of outgoing links, each of which is labeled with the probability that a unit of flow is sent down that link. (If a link in the flow network G has a flow of 0, it does not appear in G'). The edges in G' are unidirectional.

Demand nodes $d \in D$ in the original flow network G are referred to as *absorbing nodes* in G'. Associated with every non-absorbing node $v_i \in G'$ (and every link $e_i \in G'$) is a *map* data structure $v_i.map$ (or $e_i.map$) that stores the entire set of demand nodes reachable from v_i (or e_i). Each entry in this map contains an identifier of a demand node d_j, together with the probability $P(d_j|c_i)$ that a unit of flow will reach d_j from the node/link c_i to which the map belongs. Metrics can be computed by examining the contents of the map and the attributes of the relevant absorbing/demand nodes.

The method for computing conditional probabilities is shown in Algorithm 1. It proceeds by exploring secondary graph G' in reverse, tracing out paths from each absorbing node $d \in D$ to a source. For each d, a search of the graph is performed by following incoming edges $e = (src, dst, probability)$, recording the probability of arriving at d at each subsequent component.

```
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]          // amount of 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: A Deterministic Algorithm for Computing Flow Probabilities.

A lookup table containing probability values for each node is maintained in a helper variable *excess*, indexed by node ID. The *IncrementOrAddProbability*() function updates the estimate of $P(d|c)$ stored in the map of component c. The lookup table and variable *amt* are used to avoid problems with overlapping paths.

Once the relevant probabilities have been determined, the CFC of a component c_i (edge or node) is computed as:

$$C^{PCF}(c_i) = \sum_{d_j \in c_i.map} P(d_j|c_i) c_r(d_i) f_A(c_i) \tag{12}$$

where d_j is a demand node, $P(d_j|c_i)$ is the probability that flow passing through c_i reaches d_i, $f_A(c_i)$ is the flow passing through c_i under assignment A, and $c_r(d_i)$ is the criticality of d_i. The measure may be normalized by the maximum CFC value for the network. For vertices, this gives:

$$C'^{PCF}(v_i) = \frac{\sum_{d_i \in v_i.map} P(d_i|v_i) c_r(d_i) f_A(v_i)}{\max_{v \in V}(C^{PCF}(v))} \tag{13}$$

Figure 3 shows the four stages of computation. In the upper left, the basic network model is shown along with demands for each lot and each building (black font) and criticality values (white font). The upper right diagram shows a flow solution computed on the basic network, while the lower left shows the induced probability graph arising from this flow. (Network edges with zero flow are elided from the probability graph). Finally, the lower right diagram shows the critical flow metrics, ranging from green (low criticality) to red (high criticality).

Computational Requirements

Algorithm 1 is quadratic in space and time. In the worst case, the map at each node/link stores $|D|$ entries, one for each demand node in the network, leading to $O((|V|+|E|)|D|)$ in storage space. As a variant of depth first search, the time required to compute probabilities for each demand node is $O(|V| + |E|)$, so the total time required for all demand nodes is again $O((|V| + |E|)|D|)$. For infrastructure networks, $|V| \approx |E|$ and $|D| \lessapprox \frac{1}{2}|V|$, yielding time and space complexity bounds of $O(|V|^2)$. For the model used in this paper, the maps at nodes/links contain an average of 5 entries each, pushing the storage requirements closer to $O(|V|)$.

The running time of the entire method is dominated by the $O(|V||E|^2) \approx (|V|^3)$ Edmonds-Karp algorithm that is used to generate flows. If a *push-relabel* algorithm is used, the running time can be reduced to $O(|V|^2\sqrt{|E|}) \approx O(|V|^2\sqrt{|V|})$. In contrast, betweenness centrality (Freeman 1977) can be computed in $O(|V|^3)$ using the Floyd-Warshall algorithm or $O(|V|^2 \log|V| + |V||E|) \approx O(|V|^2 \log|V|)$ time using Johnson's or Brandes' algorithm [see (Cormen et al. 2009)].

Note that Algorithm 1 does not work for a transition graph G' that contains *cycles*. The Edmonds-Karp algorithm does not create cycles, but this is not necessarily the case for domain-specific methods (e.g., hydraulic simulation). For flow networks with cycles, alternative techniques must be used.

Sampling Variants

The present instantiation involves a 'one shot' scenario in which a single set of integer-valued demand values is used. $E[f_A(c_i, d_j)]$, the expected amount of flow from component c_i to demand node d_i, is computed with the use of a conditional probability $P(d_j|c_i)$. If empirical data is available, or if demand values are sampled from probability distributions, the basic method must be adapted to support sampling. This is accomplished easily with a modification that iterates over input data sets and computes $E[f_A(c_i, d_j)]$ as a sample mean of conditional probabilities.

EVALUATION

The utility of the critical flow method presented in this paper is demonstrated in two ways: (1) comparison with alternative flow centrality measures, and; (2) integration with two common forms of reliability analysis. In both cases, a synthetic, neighborhood-level model (suitable for diagrams) is used as a testbed. A model consists of a planar region containing one or more *blocks* (light purple). Each block may contain a building (dark purple) or a set of lots (light green) representing houses. A water distribution network provides buildings and lots with water drawn from a reservoir. The basic topology of the street and water network is taken from downtown Toronto, while demand values for buildings and lots are sourced from empirical studies (Aquacraft 2011). The modeler assigns criticality values to buildings manually.

Comparison with Flow Centrality Measures

The FC, SSFC, and CFC measures were computed for a small model that was patterned after high density neighbourhoods in Toronto, Canada. Figure 5 shows four views of this basic model. In the upper left, the network is shown along with the demands at each building and lot. The flow resulting from this configuration is shown in the upper right, where black network edges indicate that no flow is present. In the lower left, the SSFC centrality measure has been computed for each edge and vertex. The lower right shows the FC centrality measure.

The FC measure places emphasis on the central nodes in the graph, leaving the sole edge emanating from the source (i.e., the most critical) relatively dark. In contrast, the SSFC is highly correlated with the flow, as shown in Figure 6. The fact that both the SSFC and the maximum flow have occasional black edges is an artefact of the Edmonds-Karp algorithm that was used to produce flows, and something that would be rectified with domain-specific methods.

Figure 7 (A) shows the critical flow for the same model and demand/criticality configuration. Criticality levels for buildings are displayed in white font; lot criticality is negligible and elided. The critical flow comes from the hospital (1), school (0.6) and public buildings (0.2) down the right of the network. Figure 7 (B) arises from a single modification: the demand on one of the non-critical buildings is raised from 150 to 2000. This forces a redistribution of flows on the network due to capacity constraints on the right hand path. As a result, critical flow is more evenly balanced across all paths from the source.

The SSFC and FC measures are unable to capture the flow redistribution involved in Figure 7. To a large extent this is a result of SSFC and FC being based on sums of maximum flows between vertex subsets (i.e., pairs in the case of FC, sources and sinks in the case of SSFC). Since the underlying graph is only changing slightly between scenario (A) and (B) of Figure 7, the measures change only slightly. In contrast, CFC is computed on a per-flow basis, where the flow in question is the result of competition between demand nodes for supply. The CFC is therefore a measure of not only the network's topology, but also the *load* induced by an assignment of supply, demand, and capacity values.

Edge Reliability

The critical flow measures introduced in this paper may be combined with reliability estimation procedures [e.g., edge reliability (Zio 2007)]. As a simple example, a **reliability function** $r : E \rightarrow [0, 1]$ can be used to assign edges $e \in E$ a reliability rating $r(e)$. Edges with high criticality and low reliability can be identified by using the **Reliable Critical Flow** ("RCF") CIM:

$$C'^{RCF}(e) = C'^{PCF}(e)r(e) \tag{14}$$

Since both the normalized CFC and reliability (e) lie in the interval $[0, 1]$, the same is true for $C'^{RCF}(e)$. Reliability values can be obtained through a variety of means, including surveys, inspections, or inferences from the physical characteristics (e.g., age) of the relevant components.

Diagram A of Figure 8 shows reliability values for the model in Figure 3 . Diagram B is a plot of edge reliability [versus CFC. The majority of the edges have negligible CFC (e.g., those edges that feed individual lots). The solitary edge with a criticality of 1.0 is the edge from the reservoir that carries flow for the entire model. Of particular interest are those edges that have lower reliability ratings and moderate criticality.

While edge reliability can be combined with FC, SSFC or other centrality measures, the use of the CFC provides a snapshot of the actual critical load on the network under a given assignment. Use of the FC measure, for instance, would rank components by their role in facilitating maximum flows between all pairs of vertices, weighted by reliability. While this might be useful for social networks in which information flows arbitrarily, it does not match the typical load patterns on source/sink networks.

Component Failure Analysis

Critical flow measures also support a common form of reliability analysis in which components are failed iteratively in order to ascertain the impact on the global performance of the network [e.g., (Dunn and Wilkinson 2012)]. The total amount of critical flow (i.e., the flow delivered to demand nodes, weighted by criticality) can be used as a performance measure. Each edge/node can be failed iteratively in a 'leave one out' analysis, and the impact on total critical flow can be recorded. This type of analysis can be used to identify the portions of the network that lack fallback routes.

Figure 9 shows the result of using this approach on the edges of the network in Figure 5. The vast majority of edges carry little critical flow, and therefore have insignificant failure loss. More interesting are the edges on the upper left, with moderate CFC and no failure loss. These are edges for which there are backup routes available (e.g., the edges in the residential blocks). The edge in the upper right is the edge incident on the lone source node, the failure of which is catastrophic.

CONCLUSION

The goal of this paper was to define a novel set of *component importance measures* ("CIMs") based on critical flows and to provide a method for their computation. In the ensuing discussion, the critical flow measures were compared against two versions of the traditional flow centrality measure. A simple example of how the critical flow measures could be combined with reliability analysis was also provided.

Unlike flow centrality, critical flow measures are able to track shifts in flow distribution that arise from changes in both demand and network topology. Flow centrality aggregates the distribution of maximal flows generated between pairs of vertices, resulting in a measure that largely tracks the topology of the network, and one that is not necessarily suitable for networks with defined sets of sources and sinks. In contrast, critical flow measures are specific to a particular configuration of source capacities and sink demands. This means that they give an ephemeral glimpse of a system under load, as opposed to a more general evaluation based on network topology alone.

There are many extant works concerning the identification of critical components in infrastructure systems. The method in this paper differs in several key details. First, it allows an analyst to assign criticality values to assets such as buildings. Second, the method is agnostic about solution methods, allowing a wide range of techniques to be used. Given an assignment of demand, supply and capacity values (generated by simulation packages or obtained as empirical data) the CFC values for each network component can be computed efficiently and visualized. The use of edge failure methods can identify those components that are highly critical but lacking fallback routes.

In general, the method is intended to aid municipalities in reasoning about the consequences of interventions. For instance, it could be used in GIS software in order to help urban planners consider the impacts of zoning decisions (e.g., densification). Maintenance engineers could also use such a tool when developing schedules for the repair of infrastructure components, since: (1) highly critical components should be serviced more frequently, and; (2) repairing one set of components can shift critical flow across the network, potentially introducing new risks. Components without fallbacks might require system upgrades to provide alternative delivery paths in case of failure.

The approach shown in this paper is only a simple instantiation of the general method, providing simple means of generating flows and computing probabilities. Future work should investigate the use of domain-specific methods, as well as approaches for computing probabilities that work with cyclic graphs. Perhaps the most urgent need is to discuss the use of the critical flow method to analyze a system as it evolves over time.

There are many avenues of future work. First, performance measures can be introduced to give a metric that enables comparison and optimization. The use of utility functions was suggested, but not developed within the paper. Second, vertices and sources could be given capacity constraints, and transmission nodes/edges could be augmented to act simultaneously as supply or demand nodes (e.g., edges could have a small loss to model malfunction or component degradation). Third, network clustering algorithms [see (Brandes and Erlebach 2005)] could be introduced to reduce the number of demand nodes. Lastly, no attempt was made to identify groups of critical components (e.g., critical paths or clusters).

Data Availability Statement

Some data, models, or code generated or used during the study are available from the corresponding author by request: (1) ESRI CityEngine lot/parcel geometry and network topology in XML format; (2) Python scripts to extract lot/parcel geometry and network topology from CityEngine; (3) CityEngine scene file for the sample city used in the paper, and; (4) C++ source code for the probability propagation algorithm.

REFERENCES

Agryzkov, T., Tortosa, L., and Vicent, J. F. (2019). “A variant of the current flow betweenness centrality and its application in urban networks.” *Applied Mathematics and Computation*, 347(April), 600–615.

Ahuja, R., Magnanti, T., and Orlin, J. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ.

Almoghathawi, Y. and Barker, K. (2019). “Component importance measures for interdependent infrastructure network resilience.” *Computers and Industrial Engineering*, 133(August 2017), 153–164.

Amin, M. (2007). “Toward Secure and Resilient Interdependent Infrastructures.” *Journal of Infrastructure Systems*, 8(3), 67–75.

Apostolakis, G. E. and Lemon, D. M. (2005). “A screening methodology for the identification and ranking of infrastructure vulnerabilities due to terrorism.” *Risk Analysis*, 25(2), 361–376.

Aquacraft (2011). “Embedded Energy in Water Studies, Study 3: End-use Water Demand Profiles.” *Report no.*, California Public Utilities Commission, Energy Division.

Blanchard, P. and Volchenkov, D. (2011). *Random Walks and Diffusions on Graphs and Databases: An Introduction*. Springer.

Borgatti, S. P. (1995). “Centrality and AIDS.” *Connections*, 18(1), 112–114.

Borgatti, S. P. (2005). “Centrality and network flow.” *Social Networks*, 27(1), 55–71.

Borgatti, S. P. and Everett, M. G. (2006). “A Graph-theoretic Perspective on Centrality.” *Social Networks*, 28(4), 466–484.

Brail, R. K. and Klosterman, R. E. (2001). *Planning Support Systems: Integrating Geographic Information Systems, Models,and Visualization Tools*. Esri Press.

Brandes, U. and Erlebach, T. (2005). *Network Analysis: Methodological Foundations*. Springer.

Cadini, F., Zio, E., and Petrescu, C. A. (2009). “Using centrality measures to rank the importance of the components of a complex network infrastructure.” *Lecture Notes in Computer Science*, 5508, 155–167.

Cormen, T., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. MIT

Press, Cambridge, MA, 3 edition.

Dangalchev, C. (2006). "Residual closeness in networks." *Physica A: Statistical Mechanics and its Applications*, 365(2), 556–564.

DeLaurentis, D. and Crossley, W. (2005). "A taxonomy-based perspective for systems of systems design methods." *IEEE International Conference on Systems, Man and Cybernetics*, 86–91.

Dolev, S., Elovici, Y., and Puzis, R. (2010). "Routing betweenness centrality." *Journal of the ACM*, 57(4), 1–27.

Dunn, S. and Wilkinson, S. M. (2012). "Identifying Critical Components in Infrastructure Networks Using Network Topology." *Journal of Infrastructure Systems*, 19(2), 157–165.

Ferrario, E., Pedroni, N., and Zio, E. (2016). "Evaluation of the robustness of critical infrastructures by Hierarchical Graph representation, clustering and Monte Carlo simulation." *Reliability Engineering and System Safety*, 155, 78–96.

Freeman, L. (1977). "A Set of Measures of Centrality Based on Betweenness." *Sociometry*, 40(1), 35–41.

Freeman, L. C., Borgatti, S. P., and White, D. R. (1991). "Centrality in valued graphs: A measure of betweenness based on network flow." *Social Networks*, 13(2), 141–154.

Geertman, S., Ferreira, J. J., Goodspeed, R., and Stillwell, J. (2015). *Planning Support Systems and Smart Cities*. Springer.

Goh, K. I., Kahng, B., and Kim, D. (2001). "Universal Behavior of Load Distribution in Scale-Free Networks." *Physical Review Letters*, 87(27), 278701–278701–4.

Holsapple, C. (2008). "Decisions and Knowledge." *Handbook on Decision Support Systems 1: Basic Themes*, F. Burstein and C. Holsapple, eds., Springer, Chapter 1.

Izquierdo, L. R. and Hanneman, R. A. (2006). *Introduction to the formal analysis of social networks using Mathematica*. University of California, Riverside, California.

Kansal, M. L. and Kumar, A. (1995). "Reliability analysis of water distribution systems under uncertainty." *Reliability Engineering and System Safety*, 50, 51–59.

Kishewitsch, S. (2015). "Toronto's unique energy challenge." *IPPSO Facto*, 29(4).

Klemm, K., Serrano, M. Á., Eguíluz, V. M., and Miguel, M. S. (2012). "A measure of individual role in collective dynamics." *Scientific Reports*, 2, 1–8.

Latora, V. and Marchiori, M. (2007). "A measure of centrality based on network efficiency." *New Journal of Physics*, 9.

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

Lü, L., Chen, D., Ren, X.-l., and Zhang, Q.-m. (2016). "Vital nodes identification in complex networks." *Physics Reports*, 650, 1–63.

Michaud, D. and Apostolakis, G. E. (2006). "Methodology for Ranking the Elements of Water-Supply Networks." *Journal of Infrastructure Systems*, 12(4), 230–242.

Murray, A. T., Matisziw, T. C., and Grebesic, T. H. (2003). "A Methodological Overview of Network Vulnerability Analysis." *Growth and Change*, 39(4), 573–592.

Nagurney, A. and Qiang, Q. (2009). *Fragile Networks: Identifying Vulnerabilities and Synergies in an Uncertain World*. John Wiley & Sons, Hoboken, NJ.

Newman, M. (2018). *Networks: An Introduction*. Oxford University Press, 2 edition.

Newman, M. E. (2001). "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality." *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 64(1), 7.

Newman, M. E. (2005). "A Measure of Betweenness Centrality based on Random Walks." *Social Networks*, 27(1), 39–54.

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

Noh, J. D. and Rieger, H. (2004). "Random Walks on Complex Networks." *Physical Review Letters*, 92(11), 118701.

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

Ouyang, M. (2013). "Comparisons of purely topological model, betweenness based model and direct current power flow model to analyze power grid vulnerability." *Chaos*, 23(2).

Ouyang, M., Zhao, L., Hong, L., and Pan, Z. (2014). "Comparisons of complex network based models and real train flow model to analyze Chinese railway vulnerability." *Reliability Engineering and System Safety*, 123, 38–46.

Pferschy, U. and Schauer, J. (2013). "The maximum flow problem with disjunctive constraints." *Journal of Combinatorial Optimization*, 26(1), 109–119.

Piraveenan, M., Prokopenko, M., and Hossain, L. (2013). "Percolation Centrality: Quantifying Graph-Theoretic Impact of Nodes during Percolation in Networks." *PLoS ONE*, 8(1).

Rinaldi, S. M., Peerenboom, J. P., and Kelly, T. K. (2001). "Identifying, understanding, and analyzing critical infrastructure interdependencies." *IEEE Control Systems Magazine*, 21(6), 11–25.

Scott, J. P. (2000). *Social Network Analysis: A Handbook*. SAGE Publications, London, UK.

Smil, V. (2019). *Growth From Microorganisms to Megacities*. MIT Press.

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

Stewart, W. J. (1994). *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press.

Stewart, W. J. (2009). *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Monitoring*. Princeton University Press.

Stride, P. (2016). "Super sewer: an introduction to the Thames Tideway tunnel project in London." *Proceedings of the Institution of Civil Engineers - Civil Engineering*, 169(2), 51–51.

Stride, P. (2019). *The Thames Tideway Tunnel: Preventing Another Great Stink*. The History Press.

UN-Habitat (2009). "Planning Sustainable Cities—Global Report on Human Settlements 2009, <www .unhabitat.org/content.asp?typeid = 19&catid = 555&cid = 5607>.

Wang, D., Wang, H., and Zou, X. (2017). "Identifying key nodes in multilayer networks based on tensor decomposition." *Chaos*, 27(063108).

Wei, D., Deng, X., Zhang, X., Deng, Y., and Mahadevan, S. (2013). “Identifying influential nodes in weighted networks based on evidence theory.” *Physica A: Statistical Mechanics and its Applications*, 392(10), 2564–2575.

Zio, E. (2007). “From complexity science to reliability efficiency: a new way of looking at complex network systems and critical infrastructures.” *International Journal of Critical Infrastructures*, 3(3/4), 488.

Zio, E. (2013). *The Monte Carlo Simulation Method for System Reliability and Risk Analysis*. Springer, London, UK.

Zio, E. and Kroger, W. (2009). “Vulnerability assessment of critical infrastructures.” *IEEE Reliability Society*, 1–7.

List of Figures

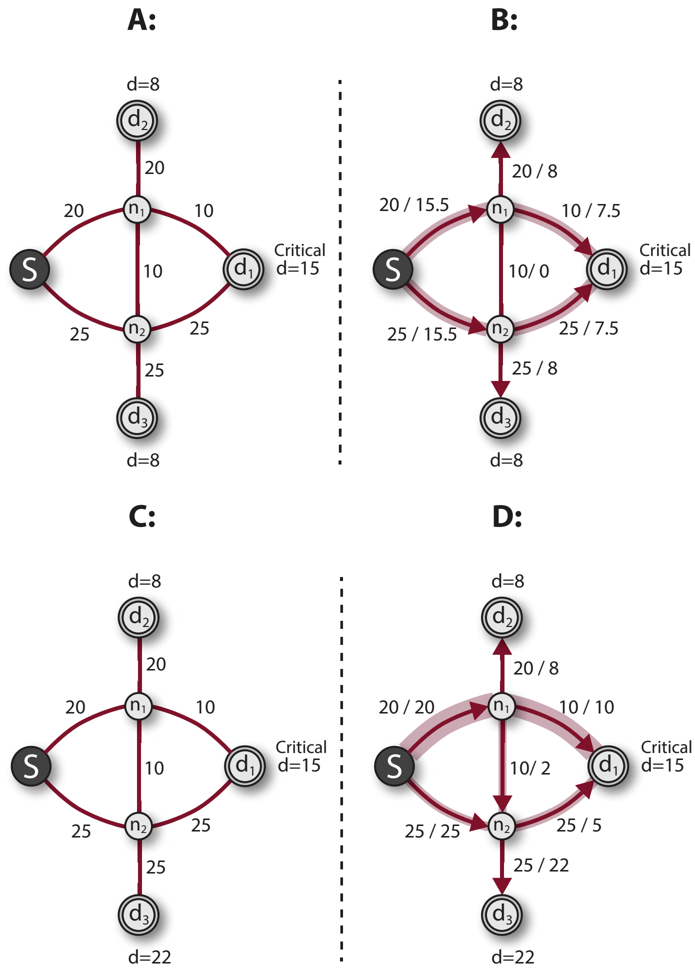

Fig. 1. A capacitated network (A) and a maximum flow solution (B) showing critical flows. Demand node d_1 is deemed a critical location. Capacities and flow values are given as X/Y where X is the capacity and Y is the flow value, and demands are indicated as $d = x$. (C) shows the same network as (A), but node d_3 has greatly increased demand. (D) shows how the upper path now carries the bulk of critical flow to d_1.

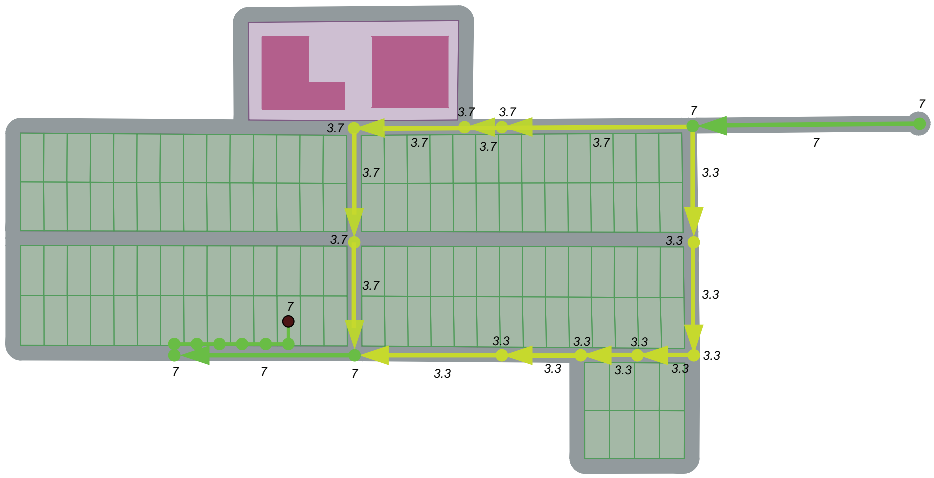

Fig. 2. Calculation of expected amount of flow delivered from source (green) to a demand node (black). Nodes/links that do not supply resources to the demand node are not shown.

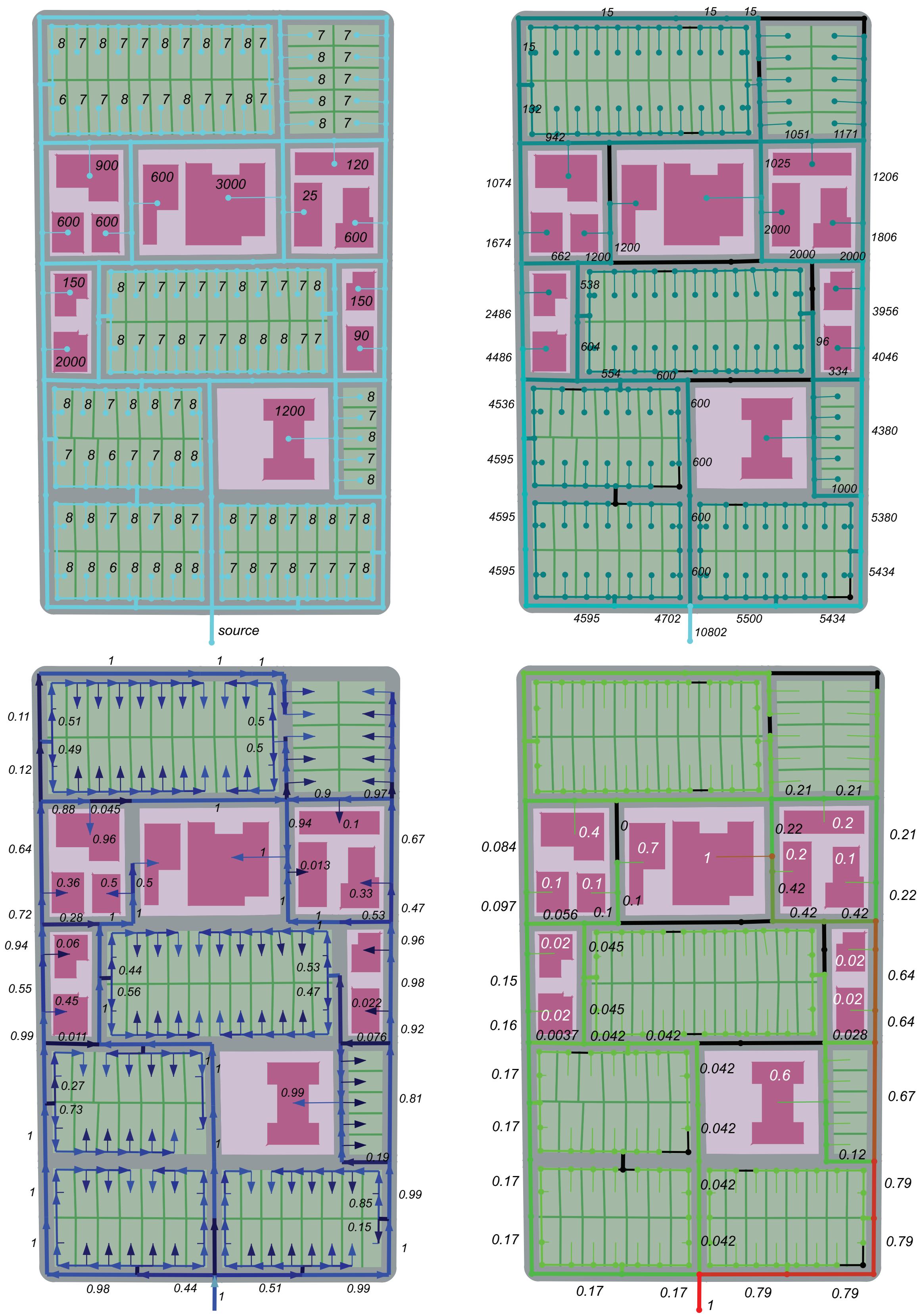

Fig. 3. Four steps in critical flow computation. Upper left: network topology with demands. Upper right: maximum flow. Lower left: induced probability graph. Lower right: critical flow (CFC), with light green being non-critical and red being critical.

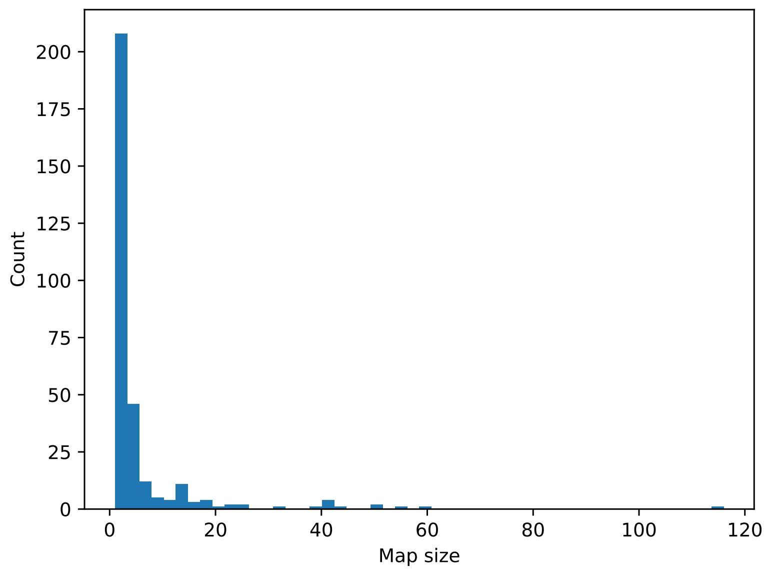

Fig. 4. Distribution of sizes for the map data structures in the model shown in Figure 3. Almost all of the nodes/links have a map with less than 6 entries.

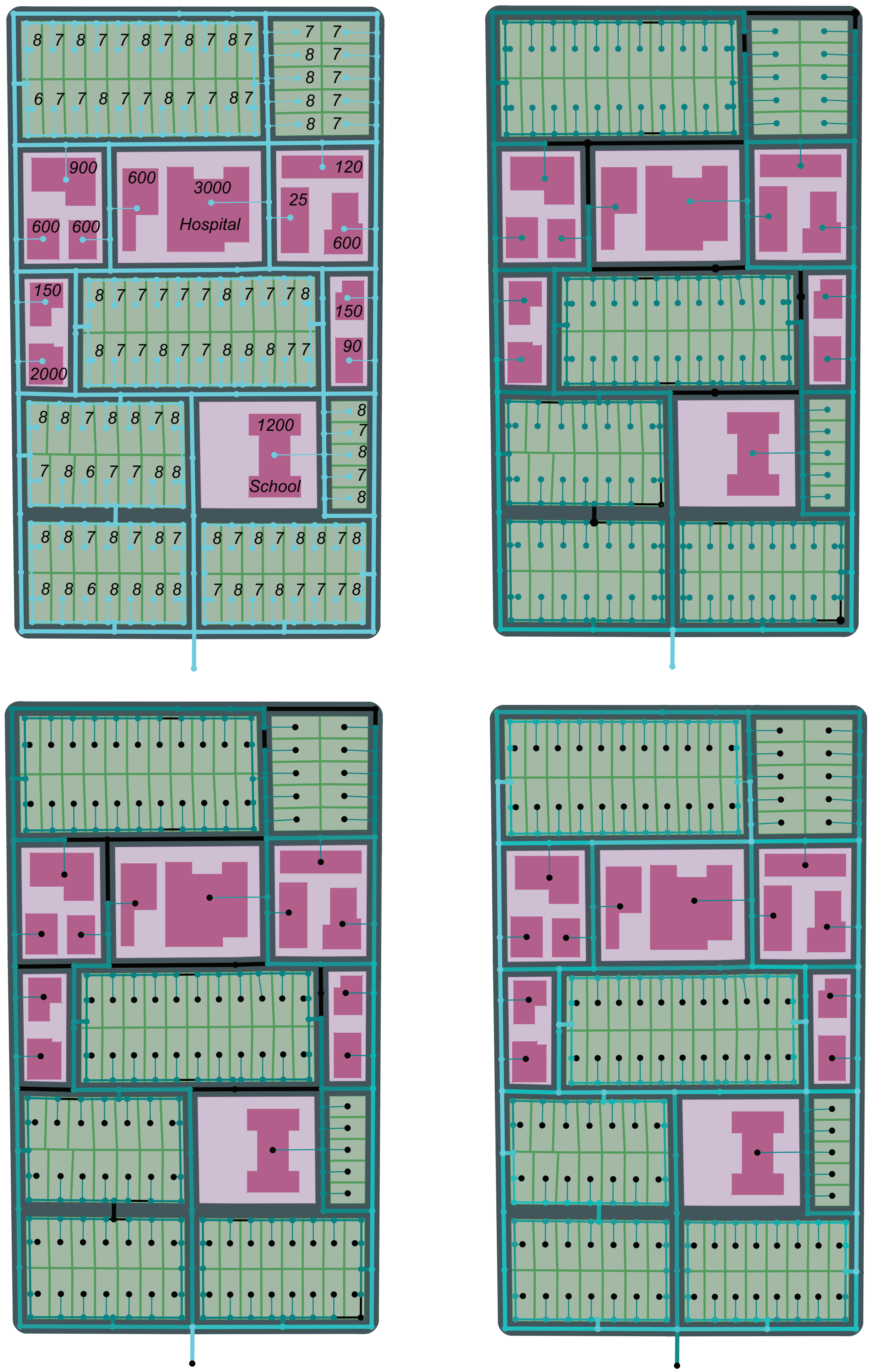

Fig. 5. Comparison of centrality measures. Upper left: empty network showing demands. Upper right: flow upon the same network. Lower left: source-sink flow centrality (SSFC). Lower right: Freeman's flow centrality (FC). The darker an edge, the lower the centrality score. (Pure black edges have no data).

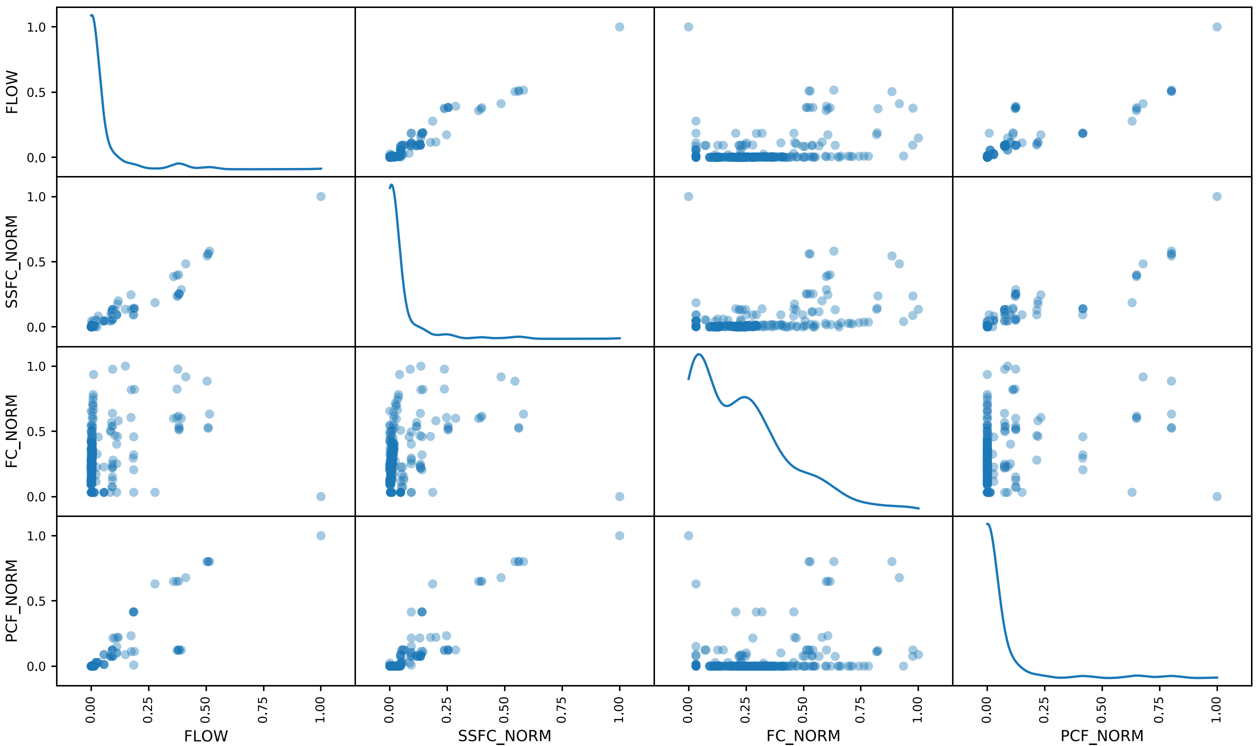

Fig. 6. Correlations between CIMs on the network in Figure 5. Normalized flow (FLOW), normalized probabilistic critical flow (PCF_NORM), normalized source-sink flow centrality (SSFC_NORM), and normalized flow centrality (FC_NORM). Diagonal elements show kernel density estimators for each CIM.

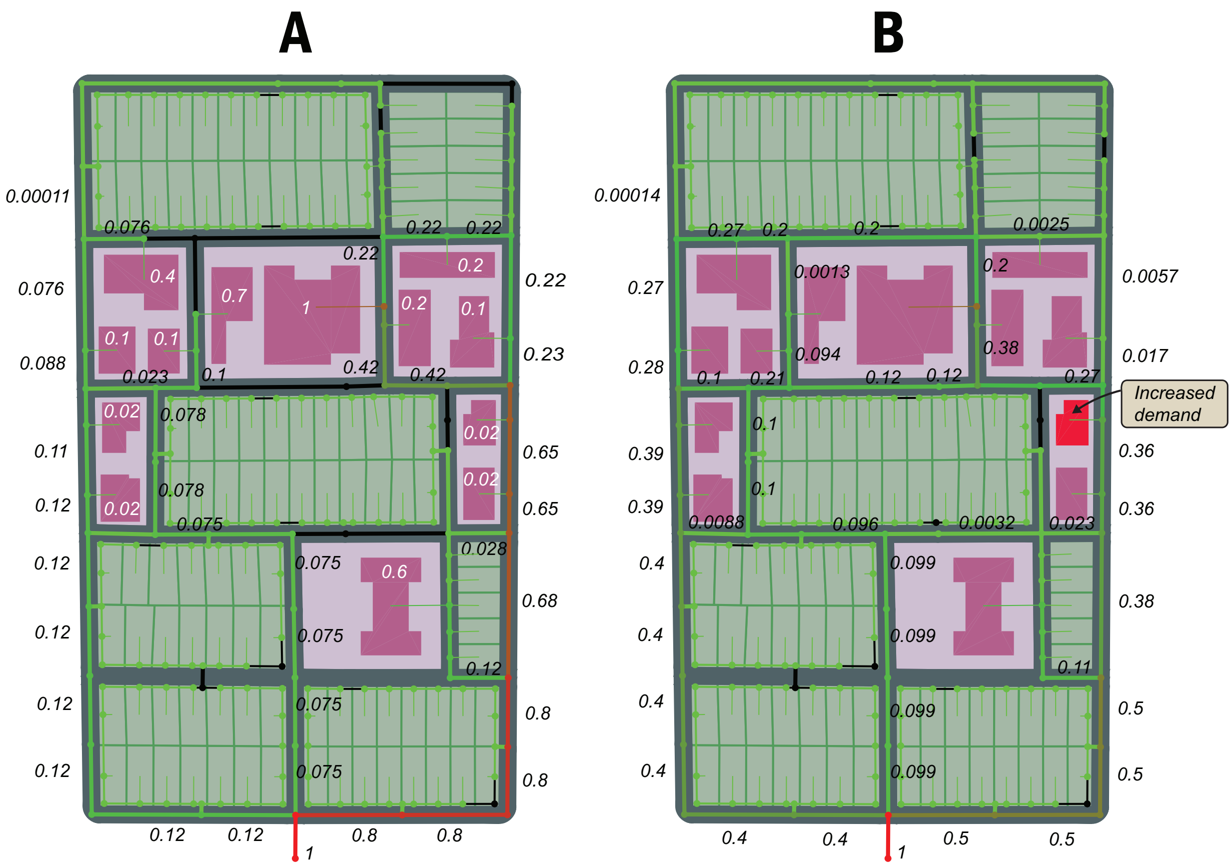

Fig. 7. CFC measures. **A**: identical model and demand/criticality configuration to that in Figure 5. Criticality levels for buildings are show in white, while lot criticality is fixed at 0.05. Critical flow favors the right side of the network. **B**: a variation in which a single building has had its demand raised from 150 to 2000. Flow patterns are changed drastically, with critical flow being redistributed across the network. Black edges have no data, light green edges are non-critical, and red edges are critical.

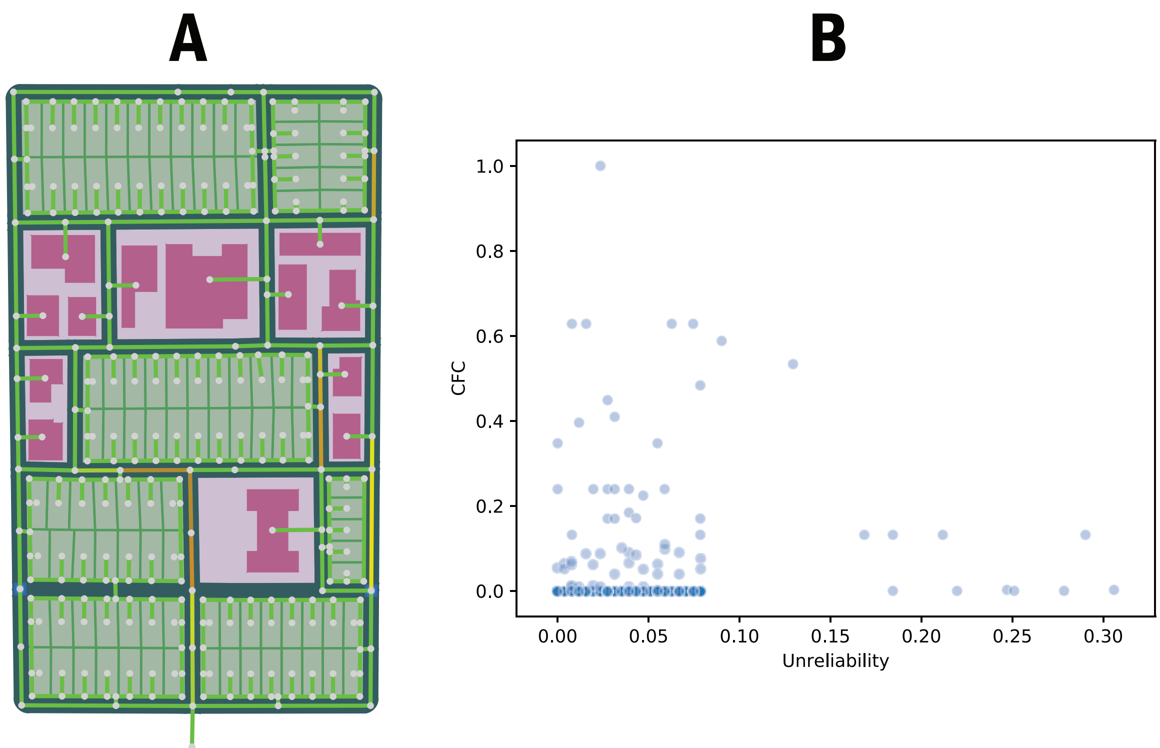

Fig. 8. Edge unreliability and critical flow. Diagram A shows edge reliability levels, ranging from green (reliable) to red (unreliable). Diagram B plots corresponding reliability values against critical flow values (taken from Diagram A of Figure 7).

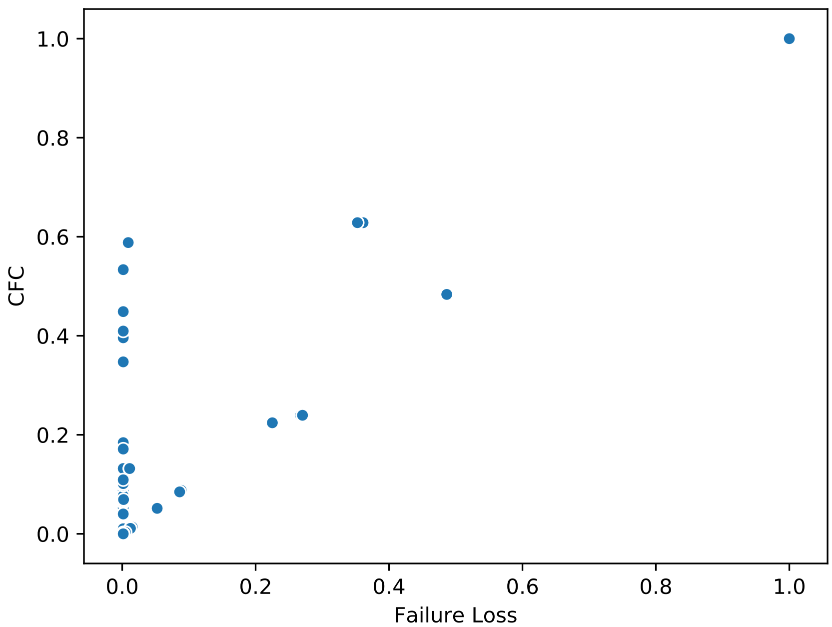

Fig. 9. CFC and failure loss values for edges.