Identifying Sensitive Components in Infrastructure Networks via Critical Flows

James Williams, jamesbw@cs.toronto.edu

April 2021

Abstract

This paper introduces a set of component importance measures that are based on the concept of critical flow. In particular, it shows how flow-based centrality measures can be used to identify critical components of an infrastructure system. The motivation of the work 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 critical flow measures may be combined with standard approaches to reliability analysis.

1 Introduction

This paper describes a set of *component importance measures* ("**CIMs**") that rank system components according to their role in the delivery of resources to critical locations. These measures can be viewed as variants of *flow centrality* [20] 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**") [24] for use by municipal governments (see [22, 11]). As urbanization continues at a rapid pace [53, 47] 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 [51, 52]).

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 entails the construction of high density residential units that impose heavier demands on legacy infrastructure systems [27]. These changes in demand patterns can altering existing flow distributions and potentially change the set of components that are relied upon to deliver resources to critical locations.

In the worst case, changes in demand 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. Resource flow to critical locations may also 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 new technique for identifying critical components of an infrastructure system, where a component is deemed *critical* according to the role it plays in providing critical locations with resources (i.e., its participation in *critical flows*). While the critical infrastructure protection literature contains methods for identifying vulnerabilities in networks containing critical assets (e.g., [5]), the present method allows criticality ratings to propagate through a network. It is also highly granular, intended for use in urban decision support systems operating at a parcel level. Furthermore, by combining critical flow with standard reliability techniques, one 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 [4]. The resulting 'systems-of-systems' [16] are subject to additional risks that arise from the complex interaction of their component systems, including *cascading failures* [45]. 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 background information on CIMs. Section Three introduces the concept of critical flow. Section Four shows how critical flow can be used to construct several new CIMs. Section Five provides a simple method for calculating probabilistic critical flow measures. Section Six provides a comparison with other centrality measures, as well as a demonstration of how reliability analysis might be integrated with critical flow modeling.

2 Background and Previous Work

Many research communities have proposed *component importance measures* ("CIMs") [3] for infrastructure systems. Examples of CIMs include network efficiency [34], flow vulnerability [42], and flow capacity rate [38]. Of particular interest are the *centrality measures*, which are used to identify the most central components in a network [48, 9, 10, 30, 37]. Arranged in categories (e.g,. [46, 42]), these include:

1. *Nearness measures*, which calculate a component's centrality by means of its proximity to other components (e.g., degree centrality [37], closeness centrality [8], residual closeness centrality [15], information centrality [29], and evidential centrality [55]).

2. *Betweenness measures*, which consider components to be central to the extent to which they stand between other components as intermediaries (e.g., shortest-path betweenness centrality [20], general-path betweenness centrality [25], load centrality [23], and random walk betweenness centrality [36]).

3. *Dynamical measures*, which examine both topology and dynamical processes situated on the network (e.g., flow centrality [21], traffic load centrality [35], random walk centrality [39, 36], routing betweenness centrality [17], dynamical influence [28], efficiency centrality [54], and percolation centrality [44]).

2.1 Flow-based Centrality Measures

Several centrality measures are based on the concept of network flow, of which the most popular is the classical formulation of *flow centrality* ("**FC**") [21]. It can be viewed as a hybrid that combines: (1) betweenness, and; (2) a representation of network dynamics. In particular, 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. FC is defined formally as follows:

Definition 1: Flow Centrality

Consider a flow network with nodes V and links E. 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** 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 measure 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* [38]). 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:

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

2.2 Flow-based CIMs Applied to Infrastructure

Various research communities have applied centrality measures to the study of infrastructure systems (see, for example, [33, 56, 13, 31]). 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., [58, 41, 42]).

Probabilistic estimates of reliability in a flow network are provided in [26]. 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 presented in [32]. 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 [18]. 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 described in [19]. 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 [57] 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. However, assigning flow based on node priority is markedly different from the method proposed in the present work.

Flow-based vulnerability measures for a capacitated and directed network are developed in [38]. 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 [2]. 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 the present paper.

Finally, a variant of *current-flow betweenness centrality* [36] was applied to to urban networks in [1]. 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 [19] permits network components to be labeled with criticality values of the sort commonly used in critical infrastructure protection (e.g., [5]). In that work, the algorithm for generating the flow solution ensures that higher priority nodes are satisfied first. In contrast, the present 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.

3 Critical Flows

This paper describes 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 *critical flows*–flows that deliver resources to critical locations such as hospitals. Components are deemed *critical* to the extent that they participate in critical flows.

For example, changes in demand can result in significant changes to the distribution of flows. 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). This can result in increased risk in cases where components in the upper path are unreliable or scheduled for maintenance.

3.1 Representing Criticality

The method presented in this work requires a modeler to assign *criticality ratings* to assets. There are at least three options for modeling asset criticality: (1) a *binary representation*, in which an asset is either critical or not; (2) a *categorical representation*, in which an asset's criticality is selected from a finite set (e.g., $\{low, med, high\}$), and; (3) a *continuous representation*, in which an asset's criticality is a real number.

After the modeler has labeled assets with criticality ratings, the rest of the system's components are evaluated. Each component is ranked according to its role in delivering resources to critical assets. There are several possible ranking methods for a component: (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 from it, and; (4) *weighted expected critical flow*, the expected amount of resource that it supplies to demand nodes, weighted by their criticality.

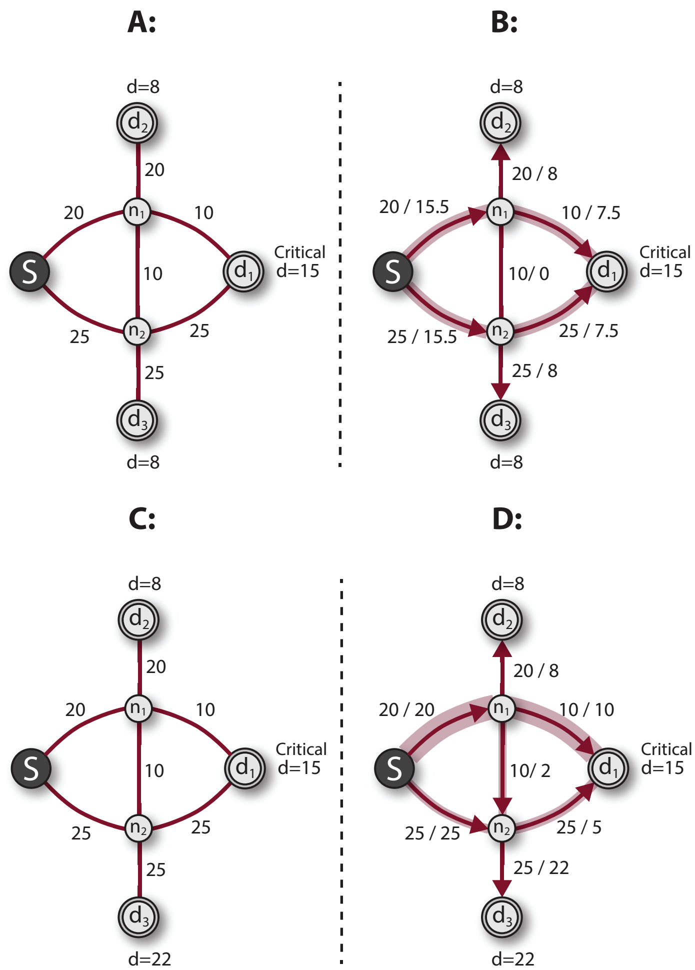

Fig. 1. A capacitated network (A) and a maximum flow solution (B) showing critical flows. Demand node d_1 is deemed a critical location. Capacities and flow values are given as X/Y where X is the capacity and Y is the flow value, and demands are indicated as $d = x$. (C) shows the same network as (A),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

4 Component Importance Measures based on Critical Flows

The concept of critical flow may be used to define numerous CIMs.

4.1 Critical Flow Centrality

The most intuitive critical flow CIM is obtained by modifying the classical notion of *flow centrality* [20]. 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.

4.2 Probabilistic Critical Flow Measures

The measures introduced above are based on maximum flows. One can relax this assumption. 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 rating c_r:

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

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:

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

This leads to the following definition (stated for vertices):

Definition 2: Vertex CFC

The (probabilistic) **critical flow centrality** (“**CFC**") of a vertex $v \in V$ is:

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

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

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

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 factors may be incorporated (e.g., reliability estimates for components, as discussed in Section 6.2). Lastly, these definitions are compatible with the use of *utility functions*, although that topic is not discussed in this paper.

5 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**").

5.1 A Method for Critical Flow Analysis

The framework for computing critical flow CIMS involves four elements:

1. a representation of an infrastructure system as a flow network;
2. input data (i.e., demand values, criticality ratings, capacity constraints);
3. a means of calculating flows, and;
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 version for clarity.

5.2 Network Representation

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 $v \in V$ has **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** $c(e) \in \mathbb{N}$ and a **flow** $f(e) \in \mathbb{N}$. 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 [14]), the explicit representation is used throughout this paper.

5.3 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** $c_r : D \rightarrow [0, 1]$ 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. These values may be provided by simulation (e.g., stochastic processes) or via external data.

5.4 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 [2], 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 [40]).

The most intuitive way to create a flow on an network is to utilize an algorithm for the MAXIMUM FLOW PROBLEM [2]. 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 [43], but they have some drawbacks). Domain-specific methods are preferable in cases where a realistic flow must be obtained. For simplicity, this paper uses the the Edmonds-Karp algorithm [see [14]].

5.5 Calculating Critical Flow from Flow Patterns

Once a flow solution has been obtained for 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 [49, 50]. Another feasible solution is to estimate probabilities from the flow network using a *Time Forward Random Walk* [7]. 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.

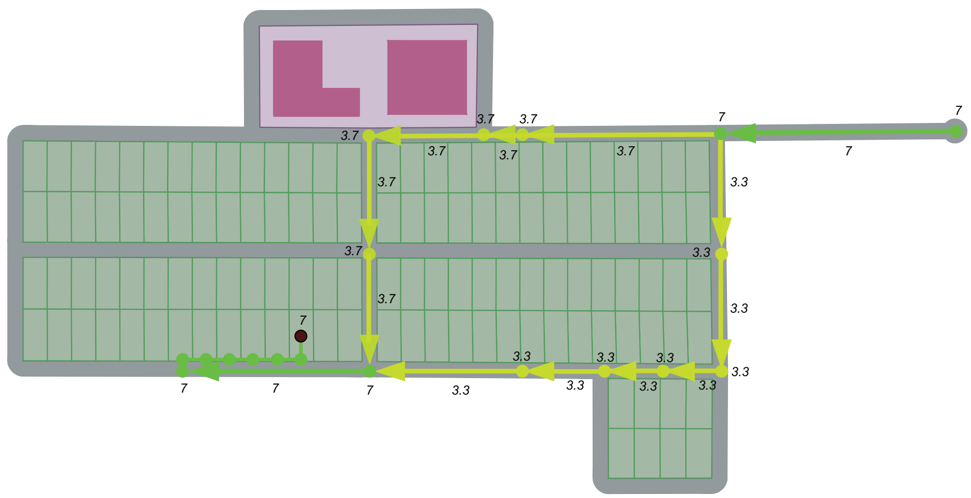

Fig. 2. Calculation of expected amount of flow delivered from source (green) to a demand node (black). Nodes/links that do not supply resources to the demand node are not shown.

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. A model consists of a planar region containing *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 [6]. The modeler assigns criticality values to buildings manually.

The basic network model is shown in the upper left, along with demands for each lot/building (black 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). Criticality ratings for buildings are in white font.

5.6 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 $O(|V|)$ (see **Figure 4**).

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 [20] 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 [14]).

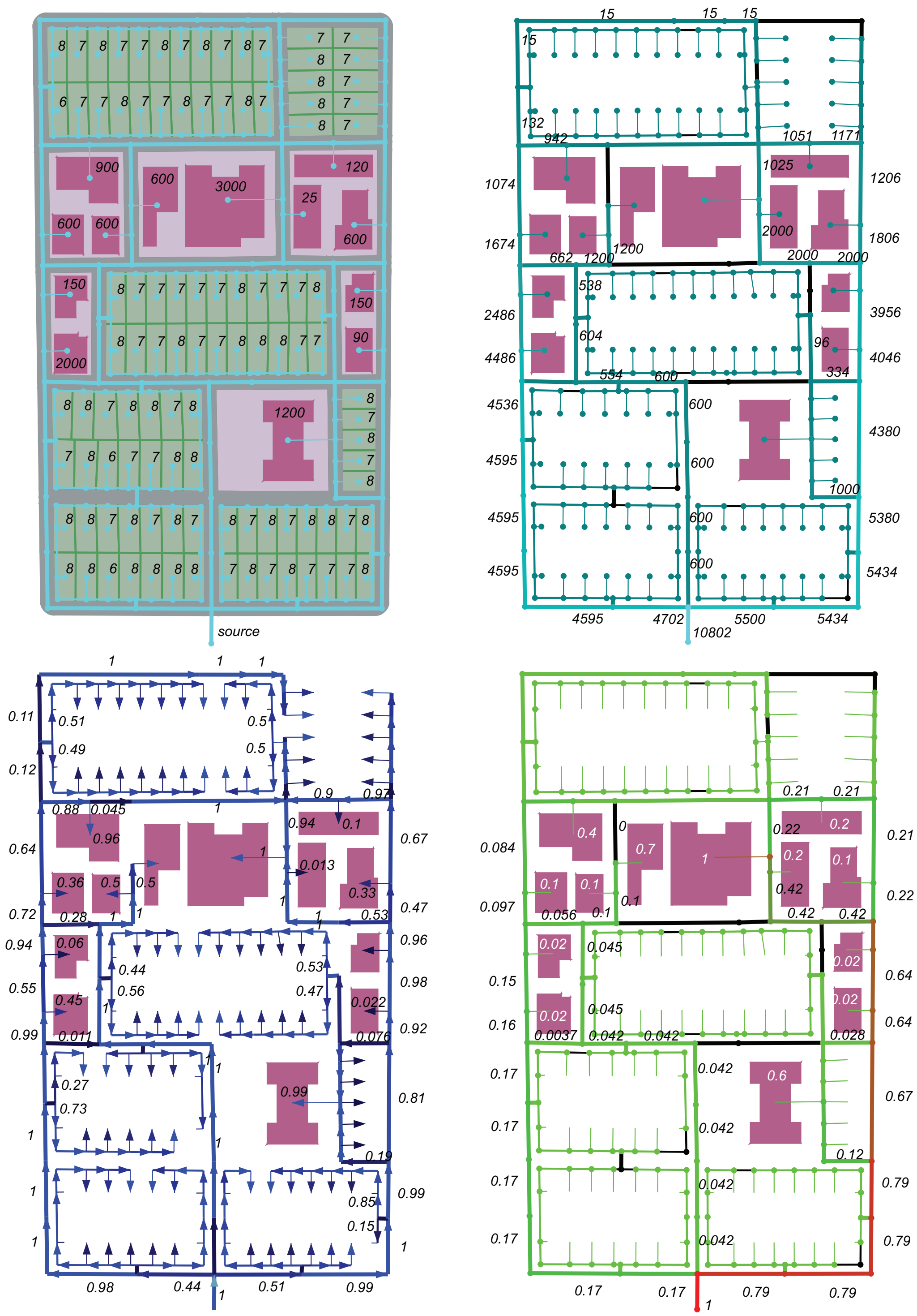

Fig. 3. Four steps in critical flow computation. Upper left: network topology with demands. Upper right: maximum flow. Lower left: induced probability graph. Lower right: critical flow (CFC), with light green being non-critical and red being critical.

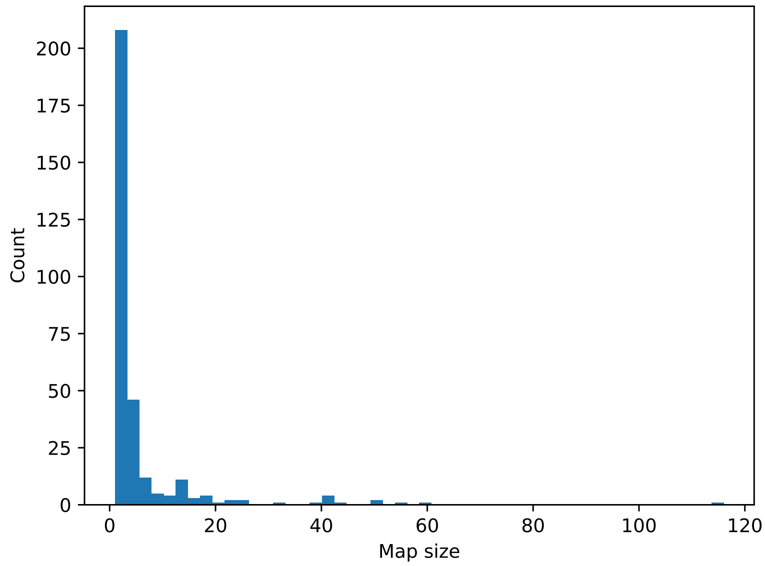

Fig. 4. Distribution of sizes for the map data structures in the model shown in Figure 3. Almost all of the nodes/links have a map with less than 6 entries.

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. For flow networks with cycles, alternative techniques must be used.

5.7 Sampling Variants

This paper presents 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.

6 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.

6.1 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. The upper left view shows the model and 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 artifact of the Edmonds-Karp algorithm.

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 subsets of V (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 SSFC and FC measures change only slightly. In contrast, the 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.

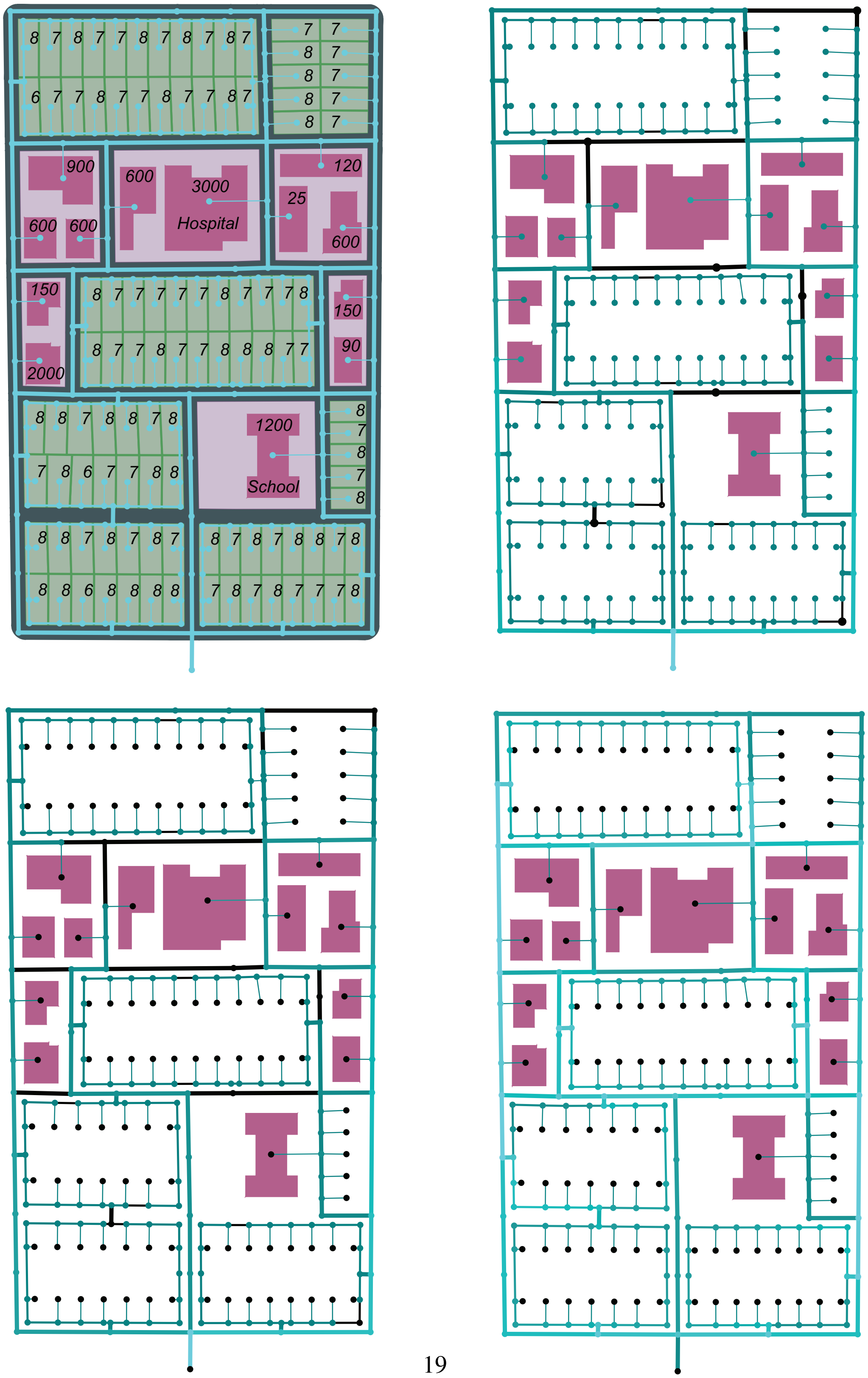

Fig. 5. Comparison of centrality measures. Upper left: empty network showing demands. Upper right: flow upon the same network. Lower left: source-sink flow centrality (SSFC). Lower right: Freeman's flow centrality (FC). The darker an edge, the lower the centrality score. (Pure black edges have no data).

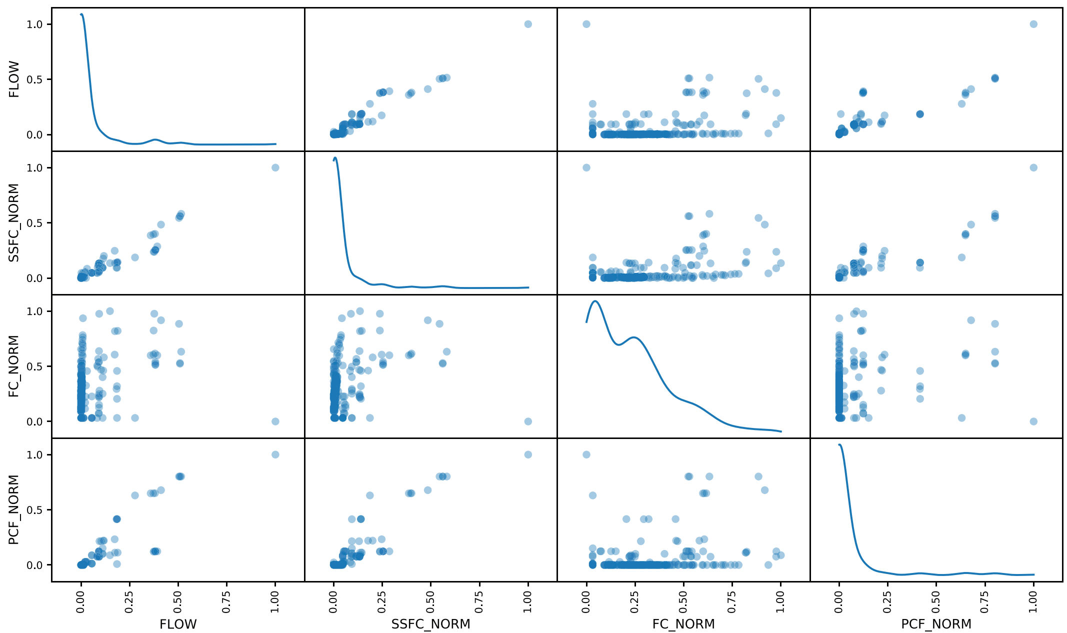

Fig. 6. Correlations between CIMs on the network in Figure 5. Normalized flow (FLOW), normalized probabilistic critical flow (PCF_NORM), normalized source-sink flow centrality (SSFC_NORM), and normalized flow centrality (FC_NORM). Diagonal elements show kernel density estimators for each CIM.

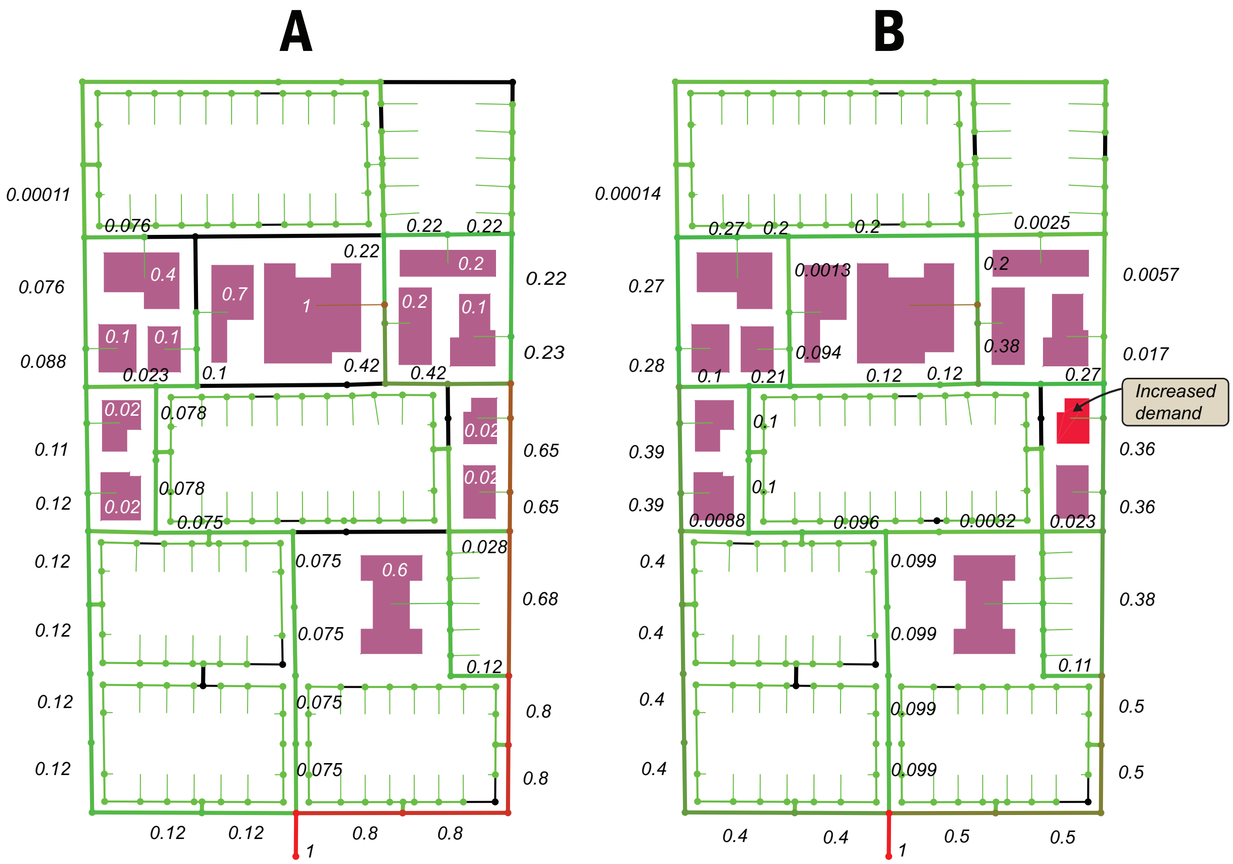

Fig. 7. CFC measures. **A**: identical model and demand/criticality configuration to that in Figure 5. Criticality levels for buildings are show in white, while lot criticality is fixed at 0.05. Critical flow favors the right side of the network. **B**: a variation in which a single building has had 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.

6.2 Edge Reliability

The critical flow measures introduced in this paper may be combined with reliability estimation procedures (e.g., edge reliability [56]). 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. Combining the FC measure with edge reliability, for instance, allows components to be ranked according to 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.

6.3 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., [18]). 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 components 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

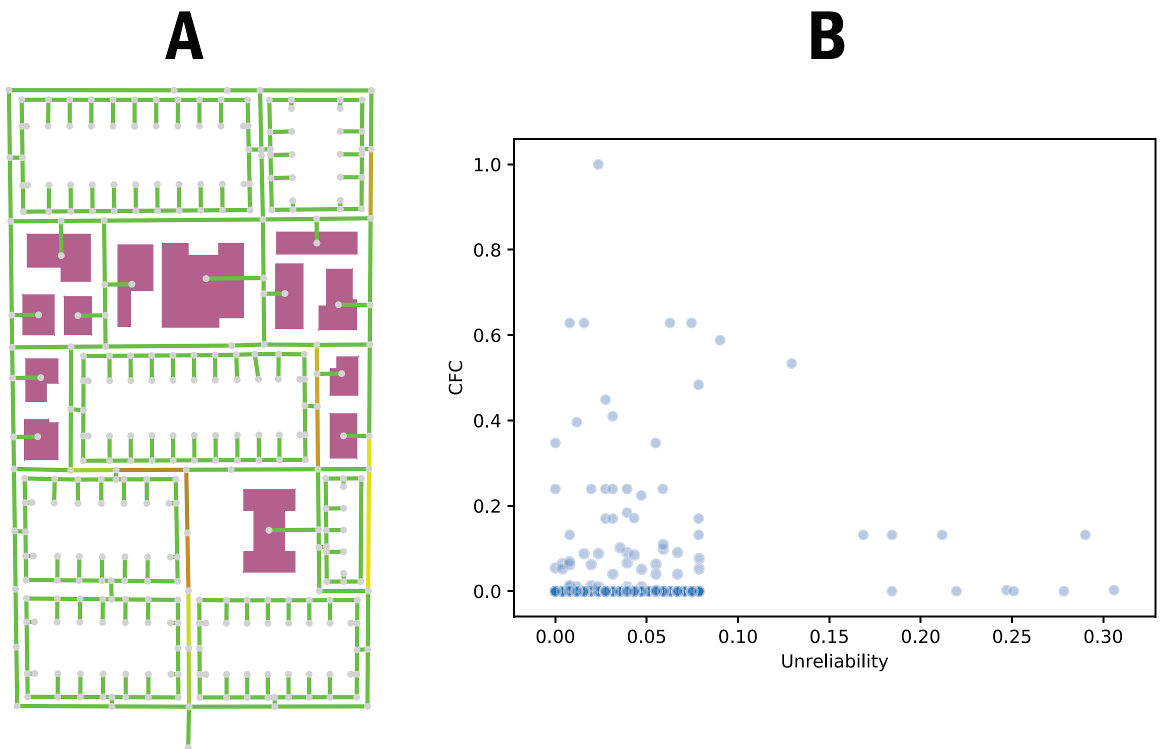

Fig. 8. Edge unreliability and critical flow. Diagram A shows edge reliability levels, ranging from green (reliable) to red (unreliable). Diagram B plots corresponding reliability values against critical flow values (taken from Diagram A of Figure 7).

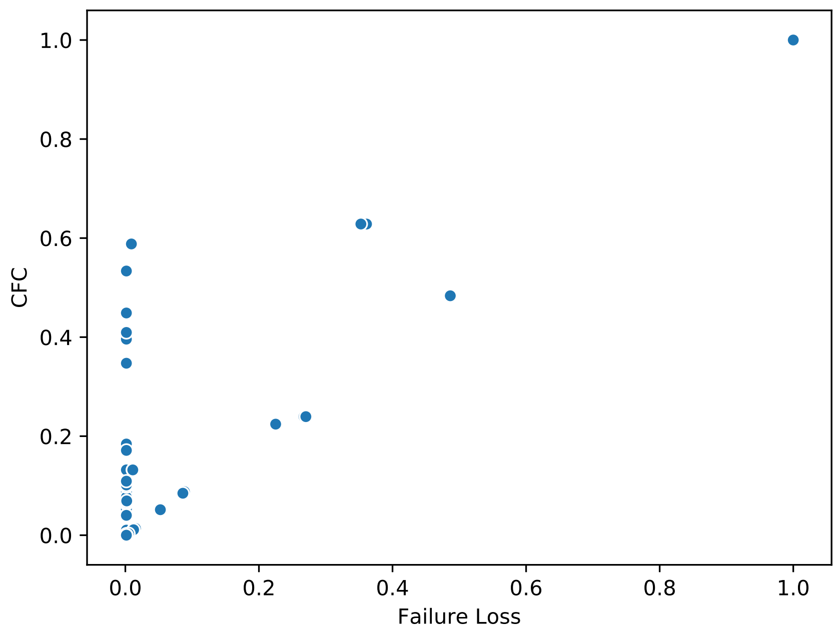

Fig. 9. CFC and failure loss values for edges.

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.

7 Conclusion

This paper introduced a set of *component importance measures* (“**CIMs**") based on the concept of critical flow. The motivation for the work was to provide urban planners and other stakeholders with a means of visualizing the impact of capacity-related changes on infrastructure systems. Various internal and external events (e.g., zoning decisions, population growth, maintenance, component degradation) can affect the distribution of flows within the system. In extreme cases, these events may interfere with the delivery of resources to critical locations such as hospitals.

The CIMs presented in this work represent an infrastructure system as a flow network in which the demand nodes are augmented with criticality ratings. Components in the system are deemed *critical* to the extent that they facilitate the delivery of resources to critical nodes. The paper showed how to use the *critical flow centrality* (“**CFC**") measure to rank components. After criticality values were assigned to the demand nodes, a probabilistic algorithm was used to propagate criticality ratings through the entire network. Although the algorithm used in the paper was discrete, a wide variety of techniques can be accommodated (e.g., simulation).

The CFC measure was compared against two versions of *flow centrality* (“**FC**") on a small, but realistic, model. The CFC was able to identify shifts in flow distribution that arise from changes in both demand and network topology. In contrast, FC aggregates maximal flows generated between pairs of vertices, resulting in a measure that largely tracks the topology of the network. The CFC is specific to a particular flow solution–that is, it represents the system under a particular load. When paired with reliability methods, it allows a modeler to identify critical nodes that lack *fallback* routes.

The methods in this paper were designed to aid municipalities in reasoning about the consequences of interventions. For instance, they 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 them 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. Finally, components that lack fallback routes require particular attention.

The approach shown in this paper is only a simple instantiation of a general method, providing a basic 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 transition graphs. Perhaps the most urgent need is to apply critical flow measures to a system that evolves over time.

There are many additional avenues for future research. First, performance measures could be introduced to yield a metric that enables comparison and optimization. Second, transmission vertices could be given capacity constraints, and transmission nodes/edges could be augmented to act simultaneously as supply or demand nodes (e.g., edges could be given a small loss to model component degradation). Third, network clustering algorithms (see [12]) could be introduced to reduce the number of demand nodes. Lastly, one could extend the method to identify groups of critical components (e.g., critical paths or clusters).

7.1 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

[1] Taras Agryzkov, Leandro Tortosa, and Jose F. Vicent. A variant of the current flow betweenness centrality and its application in urban networks. *Applied Mathematics and Computation*, 347(April):600–615, 2019.

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

[3] Yasser Almoghathawi and Kash Barker. Component importance measures for interdependent infrastructure network resilience. *Computers and Industrial Engineering*, 133(August 2017):153–164, 2019.

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

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

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

[7] Philippe Blanchard and Dimitri Volchenkov. *Random Walks and Diffusions on Graphs and Databases: An Introduction*. Springer, 2011.

[8] Stephen P Borgatti. Centrality and AIDS. *Connections*, 18(1):112–114, 1995.

[9] Stephen P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

[10] Stephen P. Borgatti and Martin G. Everett. A Graph-theoretic Perspective on Centrality. *Social Networks*, 28(4):466–484, 2006.

[11] Richard K. Brail and Richard E. Klosterman. *Planning Support Systems: Integrating Geographic Information Systems, Models,and Visualization Tools*. Esri Press, 2001.

[12] Ulrik Brandes and Thomas Erlebach. *Network Analysis: Methodological Foundations*. Springer, 2005.

[13] Francesco Cadini, Enrico Zio, and Cristina Andreea Petrescu. Using centrality measures to rank the importance of the components of a complex network infrastructure. *Lecture Notes in Computer Science*, 5508:155–167, 2009.

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

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

[16] D. DeLaurentis and W. Crossley. A taxonomy-based perspective for systems of systems design methods. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 86–91, 2005.

[17] Shlomi Dolev, Yuval Elovici, and Rami Puzis. Routing betweenness centrality. *Journal of the ACM*, 57(4):1–27, 2010.

[18] Sarah Dunn and Sean M. Wilkinson. Identifying Critical Components in Infrastructure Networks Using Network Topology. *Journal of Infrastructure Systems*, 19(2):157–165, 2012.

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

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

[21] Linton C. Freeman, Stephen P. Borgatti, and Douglas R. White. Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks*, 13(2):141–154, 1991.

[22] Stan Geertman, Joseph (Jr) Ferreira, Robert Goodspeed, and John Stillwell. *Planning Support Systems and Smart Cities*. Springer, 2015.

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

[24] Clyde Holsapple. Decisions and Knowledge. In Frada Burstein and Clyde Holsapple, editors, *Handbook on Decision Support Systems 1: Basic Themes*, chapter 1. Springer, 2008.

[25] Luis R. Izquierdo and Robert A. Hanneman. *Introduction to the formal analysis of social networks using Mathematica*. University of California, Riverside, California, 2006.

[26] M L Kansal and Arun Kumar. Reliability analysis of water distribution systems under uncertainty. *Reliability Engineering and System Safety*, 50:51–59, 1995.

[27] Stephen Kishewitsch. Toronto's unique energy challenge. *IPPSO Facto*, 29(4), 2015.

[28] Konstantin Klemm, M. Ángeles Serrano, Víctor M. Eguíluz, and Maxi San Miguel. A measure of individual role in collective dynamics. *Scientific Reports*, 2:1–8, 2012.

[29] V. Latora and M. Marchiori. A measure of centrality based on network efficiency. *New Journal of Physics*, 9, 2007.

[30] Vito Latora, Vincenzo Nicosia, and Giovanni Russo. *Complex Networks Principles , Methods and Applications*. Cambridge University Press, Cambridge (UK), 2017.

[31] Linyuan Lü, Duanbing Chen, Xiao-long Ren, and Qian-ming Zhang. Vital nodes identification in complex networks. *Physics Reports*, 650:1–63, 2016.

[32] David Michaud and George E. Apostolakis. Methodology for Ranking the Elements of Water-Supply Networks. *Journal of Infrastructure Systems*, 12(4):230–242, 2006.

[33] Alan T. Murray, Timothy C. Matisziw, and Tony H. Grebesic. A Methodological Overview of Network Vulnerability Analysis. *Growth and Change*, 39(4):573–592, 2003.

[34] Anna Nagurney and Qiang Qiang. *Fragile Networks: Identifying Vulnerabilities and Synergies in an Uncertain World*. John Wiley & Sons, Hoboken, NJ, 2009.

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

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

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

[38] Charles D. Nicholson, Kash Barker, and Jose E. Ramirez-Marquez. Flow-based vulnerability measures for network component importance: Experimentation with preparedness planning. *Reliability Engineering and System Safety*, 145:62–73, 2016.

[39] Jae Dong Noh and Heiko Rieger. Random Walks on Complex Networks. *Physical Review Letters*, 92(11):118701, 2004.

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

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

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

[43] Ulrich Pferschy and Joachim Schauer. The maximum flow problem with disjunctive constraints. *Journal of Combinatorial Optimization*, 26(1):109–119, 2013.

[44] Mahendra Piraveenan, Mikhail Prokopenko, and Liaquat Hossain. Percolation Centrality: Quantifying Graph-Theoretic Impact of Nodes during Percolation in Networks. *PLoS ONE*, 8(1), 2013.

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

[46] John P Scott. *Social Network Analysis: A Handbook*. SAGE Publications, London, UK, 2000.

[47] Vaclav Smil. *Growth From Microorganisms to Megacities*. MIT Press, 2019.

[48] Karen Stephenson and Marvin Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11(1):1–37, 1989.

[49] William J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[50] William J. Stewart. *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Monitoring*. Princeton University Press, 2009.

[51] Phil Stride. 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, 2016.

[52] Phil Stride. *The Thames Tideway Tunnel: Preventing Another Great Stink*. The History Press, 2019.

[53] UN-Habitat. Planning Sustainable Cities—Global Report on Human Settlements 2009, 2009.

[54] Dingjie Wang, Haitao Wang, and Xiufen Zou. Identifying key nodes in multilayer networks based on tensor decomposition. *Chaos*, 27(063108), 2017.

[55] Daijun Wei, Xinyang Deng, Xiaoge Zhang, Yong Deng, and Sankaran Mahadevan. Identifying influential nodes in weighted networks based on evidence theory. *Physica A: Statistical Mechanics and its Applications*, 392(10):2564–2575, 2013.

[56] 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(3/4):488, 2007.

[57] Enrico Zio. *The Monte Carlo Simulation Method for System Reliability and Risk Analysis*. Springer, London, UK, 2013.

[58] Enrico Zio and Wolfgang Kroger. Vulnerability assessment of critical infrastructures. *IEEE Reliability Society*, pages 1–7, 2009.