

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

James Bryan Williams

University of Toronto, Department of Computer Science, Canada

Abstract

This paper describes a novel method for allowing urban planners and municipal engineers to identify critical components of interdependent infrastructure networks whose attributes vary over time. The method is based on critical flow analysis, wherein system components are ranked by their role in facilitating the flow of resources to critical locations. The intent of the method is to support decision making by providing a means by which stakeholders can reason about the way in which changes in supply, demand, or network capacity can alter the distribution of critical flows within an urban environment. Individual infrastructure systems are modeled as networks that can be linked to one another by physical and geospatial dependencies. A simple instantiation of the method is presented and evaluated on a district-scale model of a city that contains water and electricity networks. The paper also discusses two forms of reliability analysis based on critical flows: a composite measure incorporating edge reliability, and a variation on standard component failure/degradation analysis.

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

1. Introduction

2 This paper presents a novel method for identifying critical components in
3 interdependent, urban infrastructure systems. The ultimate goal of the research
4 is to develop a decision support tool that allows urban planners and municipal
5 engineers to reason about risks introduced by interventions (e.g., zoning changes,
6 maintenance activities). The paper demonstrates that standard network analysis
7 techniques can be combined with criticality and reliability metrics in order to
8 define a composite method that provides useful information for decision makers.

9 Although the method described in this work can be used in a variety of contexts,
10 the paper focuses on urban infrastructure systems (excluding transport). Residents
11 of cities depend on infrastructure systems to deliver not only physical resources
12 such as water and gas, but also a range of social goods ranging from education
13 to healthcare. Disruptions in the delivery of resources and/or services can have
14 extremely deleterious consequences, particularly for critical locations such as
15 hospitals. Methods for identifying the infrastructure components that supply
16 critical locations with resources could be used in several activities, including
17 maintenance scheduling, disaster recovery, and zoning.

18 Infrastructure systems can be disrupted in numerous ways, including deliberate
19 attacks, component failures, and natural disasters. Much of the existing research
20 on critical infrastructure protection, for instance, has focused on protecting infras-
21 tructures against damage due to extreme weather or deliberate attacks [1, 2, 3].
22 Component failure has been studied extensively in the field of reliability engineer-
23 ing (e.g., [4]) and in the various engineering disciplines (e.g., water [5], drainage
24 [6], electricity [7], telecommunications [8], and transportation [9]). Disruption of
25 networks has also been considered in operations research (e.g., [10]), computer
26 science (e.g., [11, 12]), network reliability (e.g., [13, 14]), graph theory (e.g.,
27 [15, 16]), and network science (e.g., [17]).

28 While the method presented in this paper can represent disruptions, it was
29 designed to accommodate a broader set of issues. In addition to severe, short
30 term events (e.g., natural disasters), infrastructure systems are influenced by a
31 variety of factors, including: (1) *population growth*, which typically results in
32 increased demands; (2) *component degradation*, which can introduce new capacity
33 constraints; (3) *maintenance activities*, which can shift flows of resources from
34 one route to another, and; (4) *planning interventions* (e.g., the development of new
35 residential subdivisions), which can have effects both on system topology and on
36 demand patterns.

37 In order to accommodate this diverse set of scenarios, the method includes
38 three major features that, in combination, distinguish it from prior art: (1) lo-
39 cations are annotated with *criticality ratings*, allowing distinctions to be drawn
40 between different types of facility; (2) infrastructure systems may be connected
41 via geospatial and physical *dependencies*; (3) system attributes (e.g., demand for
42 resources) are modeled as *time series*, permitting the user to reason about the
43 impacts of interventions or disruptions over different time scales.

44 The structure of this paper is as follows. Section 2 provides useful background
45 information, while Section 3 introduces the methodology used in this paper. Sec-
46 tion 4 provides an evaluation of the methodology on a district-scale model of a

47 city. Section 5 discusses two forms of reliability analysis that can be combined
48 with critical flow measures. The paper closes with suggestions for future research.

49 **2. Background**

50 The method in this paper can be viewed as a combination of techniques from
51 network science and critical infrastructure protection. The fundamental building
52 block is a *component importance measure* ("CIM") (e.g., [18], [19], [20]) that
53 estimates the degree to which a given component participates in the delivery of
54 resources to critical locations. Before discussing the method in detail, a quick
55 discussion of relevant background material is required.

56 *2.1. Network Science and Centrality Measures*

57 Networks are a common choice of modeling mechanism in many fields, and
58 critical infrastructure protection is no exception (see [1]). For example, many ap-
59 proaches to infrastructure vulnerability and resilience make use of techniques from
60 network science. From the perspective of the current paper, the most important of
61 these techniques are the *centrality measures*, which are used to identify the most
62 central components in a network (see [21, 22, 23]).

63 Numerous centrality measures exist [24], the most intuitive of which are: (1)
64 *nearness measures*, which determine a given component's centrality by means
65 of its proximity to other components, and; (2) *betweenness measures*, which
66 deem components to be central to the extent to which they stand between other
67 components as intermediaries. These categories contain measures that largely
68 focus on network topology; in contrast, *dynamical measures* take into account
69 various dynamical processes taking place on the network.

70 The progenitor of the method used in this paper is *flow centrality* [25]. Consider
71 a simple network with nodes V and links E . A node v is considered to be *between*
72 other nodes u and w to the extent that the maximum flow between u and w depends
73 on v . Nodes are deemed central to the extent that they facilitate maximum flow.

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

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

74 *2.2. Interdependent Infrastructures*

75 Infrastructure systems are typically coupled to the extent that the failure of
76 components in one system can cause failures in connected systems [26]. These
77 *interdependent* systems are typically more fragile than solitary systems [27], with
78 additional failure modes (e.g., *cascading failures* [28]) that can be quite complex.
79 For example, water distribution systems impose much greater cascading damage
80 on other systems than they receive in return [29], and they seem to display a greater
81 propensity to initiate cascading failure in other systems [30].

82 Various research communities have advocated an integrated view of infras-
83 tructure systems, and a growing body of work is available on interdependencies
84 (e.g., [31, 32]). For instance, homeland security initiatives following the Septem-
85 ber 11th terrorist attacks in the United States spurred numerous efforts addressing
86 infrastructure interdependencies (e.g., [33]). Overviews of techniques for the mod-
87 eling and simulating interdependent critical infrastructure systems may be found
88 in several places, including [34].

89 *2.3. Modeling Interdependent Infrastructures with Networks*

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

95 To be precise, a network A is **dependent** on network B if the state of B can
96 influence the state of A [41] (see also [42]). Dependencies can be classified as
97 follows [30]:¹

- 98 1. **Physical dependencies**, in which the state of A is affected by the material
99 outputs/flows of B
- 100 2. **Geospatial dependencies**, in which certain components of A and B are in
101 such close spatial proximity such that local events can affect both networks;
- 102 3. **Informational dependencies**, in which A and B are connected by *informa-*
103 *tion and communications technology* ("ICT");
- 104 4. **Social dependencies**, in which A affects B along social dimensions;
- 105 5. **Procedural dependencies**, where A affects B on the basis of organizational
106 or regulatory structures, and;

¹Alternative classifications appear in [43, 44].

107 **6. Financial dependencies**, where market conditions, financial crises and other
108 economic events allow one network to affect another.

109 There are many ways to represent these dependencies in network models, a dis-
110 cussion of which is beyond the scope of the paper.

111 2.4. *Finding Critical Components in Interdependent Networks*

112 Numerous researchers have proposed methods for identifying critical compo-
113 nents in interdependent networks. Typical examples are described below:

- 114 • Apostolakis and Lemon [45] evaluate the vulnerability of geospatially inter-
115 dependent infrastructure systems (gas, water, electric) by identifying **critical**
116 **locations** — geographical points that are susceptible to attack. Each system
117 is represented as a directed network in which vertices can represent not just
118 junctions but also physical features (e.g., manhole covers). Co-location of
119 assets (e.g., shared service tunnels) is modeled by allowing vertices from
120 one graph to appear in another. (Physical dependencies, such as the use of
121 electricity by the water system, are not modeled).

122 In their approach, a set of attack scenarios is identified and the networks
123 are analyzed in order to identify *minimal cut sets* (see [7]). The resulting
124 vulnerabilities are prioritized by: (1) the degree to which the targets are
125 accessible to the attacker (i.e., susceptibility), and; (2) the value of the
126 targets from the standpoint of the decision-maker, calculated by summing
127 their expected disutilities. The susceptibility and value are combined to
128 yield a **vulnerability category** — one of five colors ranging from green to
129 red.

- 130 • Lee et al. [43] provide a method for prioritizing service restoration activities
131 in an interdependent system-of-systems. Each independent system is repre-
132 sented as a flow network that carries commodities, composed of edges and
133 vertices that may both have capacity constraints. Dependencies are modeled
134 as additional constraints in a mixed integer network flow model. In addition
135 to geospatial and physical dependencies, they allow *shared dependencies*
136 (i.e., for multi-commodity flow networks) and *exclusive-or dependencies*
137 (i.e., to allow flow on a multi-commodity network to be restricted to one
138 type of commodity at a time).

- 139 • Duenas-Osorio et al. [46] study the interdependency of electricity and
140 water systems from a topological standpoint. Both geospatial and physical

141 dependencies are modeled, with the water system requiring electricity for
142 pumps, lift stations, and control units. Conditional probability distributions
143 are used to model potential failures of water system components given
144 failure of electricity system components. Three types of vertex removal
145 strategies are used to model disruptions; for each such disruption, a set of
146 metrics are calculated: (1) nodal degree; (2) characteristic path length [47];
147 (3) clustering coefficient [48], and; redundancy ratio. Flows of water or
148 electricity are not modeled.

149 • Buldyrev et al. [26] examine the impact of electricity system disruptions on
150 the internet. Geospatial dependencies are modeled by assigning each internet
151 server to the closest power station. Disruptions are initiated by removing
152 power stations and tracking resulting nodal failures — in particular, a node
153 v is considered to be failed if: (1) all of v 's neighboring nodes are failed, or;
154 (2) the geospatially coupled node in the electricity network is failed. Nodes
155 are ranked according to the consequences of removal. The authors argue that
156 disruption of a small number of nodes in the electricity system is sufficient
157 to provide cascading failures in the internet network.

158 • Galvan and Agarwal [49] perform vulnerability analysis on interdependent
159 infrastructures by examining the impact of disruptions. Each infrastructure
160 is represented as a flow network with a unique resource type. In each
161 iteration of the analysis, a single node is selected for failure (disruption).
162 After recomputing the flow solution, the algorithm identifies every node that
163 is in violation of capacity constraints. These latter nodes are then disabled
164 and the process repeats itself until no more failures occur.

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

170 • Svendsen and Wolthusen examine interdependent critical infrastructures
171 in a series of papers [50, 51, 52, 53]. Their models represent multiple
172 concurrent types of dependencies, categorized at a high level into *storable*
173 and *non-storable* types. Each vertex v in a network can act as a producer or
174 consumer of up to m different resources, and for each such resource v has
175 a corresponding buffer. The authors investigate numerous issues, including
176 the behaviour of systems with cyclic interdependencies.

177 3. Methodology

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

183 3.1. Overview

184 The paper provides an example of such a method, based on a centrality measure
185 that combines classical flow centrality [25] with concepts from critical infrastruc-
186 ture systems (e.g., [45]). The perspective in the paper is *resource-based*, focusing
187 on the routes by which resources are delivered to consumers. Components are
188 deemed critical to the extent that they are involved in facilitating the flow of
189 resources to critical locations.

190 Computation of the centrality measure, *critical flow centrality* (“CFC”), can
191 be accomplished in several ways (see [54]). In the current paper, a discrete-valued
192 approach is taken in which: (1) an infrastructure system is represented as a flow-
193 network; (2) demands, capacities, and supply limits are given as integers, and;
194 (3) each demand node in the network is assigned a real-valued criticality rating.
195 Network flows are simulated with a standard maximum flow algorithm; once a
196 flow has been defined, a search-based algorithm computes expected contribution
197 of each component to the critical flow within the network.

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

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

209 The main thrust of the demonstration is to show that: (1) the computation
210 of CFC values can be performed efficiently, enabling their use in interactive GIS
211 applications; (2) CFC values can correctly propagate between system models,

212 and; (3) CFC computations can be integrated with standard reliability measures to
213 provide a composite view of a system.

214 The CFC measure itself is completely general, requiring only a flow solution
215 and a network topology. The method presented in this paper uses the same (dis-
216 crete) algorithms to compute values for each individual infrastructure system –
217 namely, (1) an integer-valued maximum-flow algorithm to approximate resource
218 flow within infrastructures, and; (2) a modified graph-search algorithm to compute
219 CFC values. These design choices are for ease of explanation, and more sophis-
220 ticated, heterogeneous systems can be accommodated. One can model a water
221 system using hydraulic techniques [55], for example, coupling it to an electricity
222 system that is simulated using its own domain-specific methods. Given a flow so-
223 lution and network topology, CFC values can be computed by using Markov-chain
224 Monte Carlo or random walks (see [54] for details).

225 3.1.1. Integration with GIS

226 This work was motivated by the problem of providing adequate decision sup-
227 port for urban planning. For instance, densification of urban areas is accompanied
228 by greater demand for resources; the increased demand could: (1) violate capacity
229 constraints, as in the case of the London sewer systems [56, 57], or; (2) threaten
230 the ability of a legacy infrastructure system to reliably deliver services to critical
231 locations such as hospitals and transportation hubs. Urban planners could benefit
232 from tools that allow them to visualize the impacts of land-use decisions on the
233 provision of critical resources and/or services.

234 Effective modeling of integrated infrastructure systems requires more than a
235 static, single-perspective approach. Management of disruption (and prevention of
236 cascading failures) requires an understanding of system dynamics [58]. Further-
237 more, any model used to study the disruption of interdependent infrastructures
238 needs to support two different perspectives [43]: (1) a ‘system-of-systems’ view
239 that focuses on dependencies, and; (2) a traditional view of each individual system
240 that is familiar to managers/specialists.

241 One means of providing infrastructure models that support multiple perspec-
242 tives is through the use of *geographical information systems* (“GIS”) software.
243 In fact, the critical information protection community has begun to use GIS as a
244 platform for resilience and vulnerability analysis [59]. For this reason, the method
245 described in this paper was explicitly designed for integration within GIS software.

246 *3.1.2. Data Sources*

247 Two major challenges arise when data sources are considered. First, data on
248 infrastructure systems does not always exist, and particularly not in a form that
249 permits detailed analysis of interdependencies. Second, infrastructure systems
250 in many countries (e.g., the United States power grid) are not under the control
251 of a single entity [58], making the data collection process difficult. The lack of
252 information on infrastructure assets has motivated some researchers to develop
253 techniques for inferring asset locations from proxy data sources (e.g., [60, 61]).

254 The model used in this paper is a mixture of synthetic and empirical com-
255 ponents. The basic topology (i.e., road and parcel structure) was taken from
256 downtown Toronto, albeit the boundaries were simplified in order to make dia-
257 grams feasible and to convey the basic method clearly. Resource demand profiles
258 (e.g., hourly water consumption for hospitals) were taken from empirical studies
259 and from municipal utilities.

260 *3.1.3. Implementation*

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

269 *3.2. Modeling Approach*

270 This section discusses the building blocks of the simplified model, includ-
271 ing: (1) the network representation; (2) time series representation of supply and
272 demand; (3) criticality ratings, and; (4) inter-system dependencies.

273 *3.3. Network Representation*

274 An individual infrastructure system is modeled as a weighted, capacitated, flow
275 network $G = \langle V, E \rangle$ where G is a set of nodes, $E \subseteq V \times V$ is a set of edges:

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

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

283 A network G contains both source (supply) and sink (demand) nodes. The
 284 set of **source nodes** is $V_S = \{s_1, s_2, \dots, s_p\} \subseteq V$, and the set of **demand nodes**
 285 is $V_D = \{d_1, d_2, \dots, d_k\} \subseteq V$. All other nodes are called *transmission nodes*.
 286 Multi-functional nodes are supported using a standard maximum flow reduction
 287 (as described in Section 3.8.1).

288 A **flow** on G is a real-valued function $f : E \rightarrow \mathbb{R}$ on G 's edges that obeys three
 289 flow properties:

- 290 1. **Capacity Constraints:** for all $e = (v_i, v_j) \in E$, we have $f(e) \leq c(e)$.
- 291 2. **Skew Symmetry:** for all $e = (v_i, v_j) \in E$, we have $f((v_i, v_j)) = -f((v_j, v_i))$.
- 292 3. **Flow Conservation:** for all transmission nodes $v_t \in V - (V_D \cup V_S)$, we have
 293 $\sum_{v \in V} f((v_t, v)) = 0$.

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

297 3.4. Supply Constraints and Demand Distributions

298 Supply constraints and resource demands are represented as discrete, integer-
 299 valued *time series* (see [62]). (While capacities can also be represented as time
 300 series, the demonstration assumes node and edge capacities are static.) For simplic-
 301 ity, each time series is assumed to be regularly sampled at times $t_i \in T = [0, \infty]$.
 302 They can be interpreted as the output of functions:

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

308 An **assignment** to a network involves specifying functions (time series) for all rele-
 309 vant nodes. Computations on the network (e.g., network flow solutions, criticality
 310 measures) are performed for each time $t_i \in T$. Values from previous time steps
 311 t_k may be used as input for computing values in the current time step t_i (where
 312 $t_k < t_i$). This permits the method to represent *delays* in resource utilization.

313 **3.5. Criticality Ratings**

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

318 **3.6. Interdependencies**

319 A *system-of-systems* (“SoS”) model consists of a set of k infrastructure systems
 320 $\mathcal{S} = \{S_1, S_2, \dots, S_k\}$. As shown in **Figure 1**, two types of dependencies are
 321 permitted between pairs of elements from \mathcal{S} :

- 322 1. **geospatial dependencies**, which arise when elements from network A are
 323 *co-located* with those from network B .
- 324 2. **physical dependencies**, wherein elements in network A require resources
 325 flowing through network B .

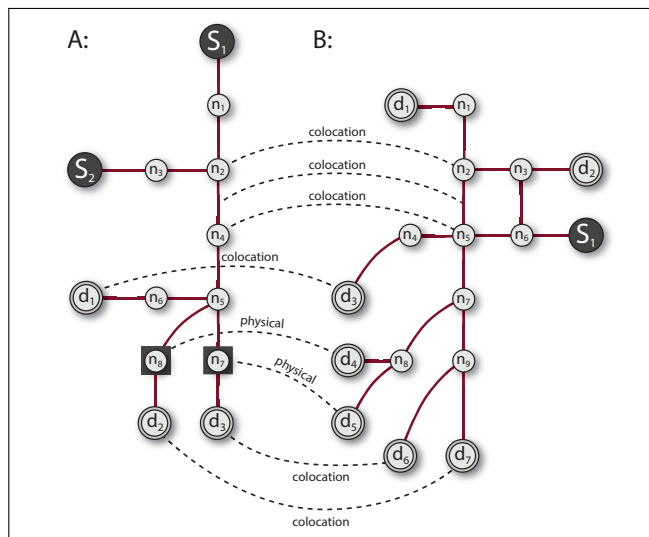


Fig. 1. Two independent infrastructure systems (A) and (B), linked by geospatial and physical (e.g., flow-related) dependencies.

326 Dependencies are represented as *interlinks* between individual infrastructure
 327 networks [35]. In contrast to [45], nodes from one network S_i do not appear
 328 directly in another network S_j . This design choice makes it easier to integrate
 329 disparate modeling methods for each individual infrastructure system (see [63]).

330 Interlinks representing physical dependencies are implemented with the use of
 331 *interconnection records*. Referring to **Figure 2**, let S_1 represent a water distribution
 332 system, and let S_2 represent an electricity system. A dependency between water
 333 node $v_1 \in S_1$ and electricity node $v_2 \in S_2$ is represented by an **interconnection**
 334 **record** $IR(v_1, v_2)$. The amount of resource R demanded of S_2 by v_1 (e.g., the
 335 amount of electricity required to operate a given water pump) is given by a function
 336 $f^R : S_1.V \rightarrow \mathbb{R}$. For instance, a pump at v_1 might demand a constant amount of
 337 electricity per unit time, or it may require power proportional to the flow $f(v_1)$
 338 through v_1 (e.g., $f^R(v) = cf(v)$). *Delays* can be accommodated by deferring this
 339 demand to later time steps.

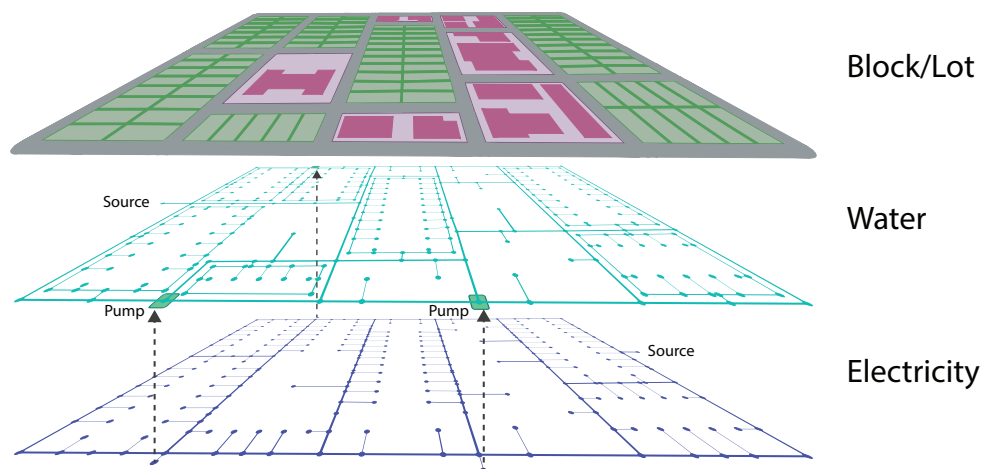


Fig. 2. Infrastructure model with two layers, showing resource flow between water pumps and electricity nodes.

340 Dependencies between network elements imply dependencies between systems.
 341 If an interconnection record exists that maps elements of S_1 to elements
 342 of S_2 , we say that S_1 is **physically dependent** on S_2 , represented as $S_1 \rightarrow S_2$.
 343 Mutual dependency between systems makes the computational task more diffi-
 344 cult. The methods of Svendsen and Wolthusen (e.g., [52]) accommodate mutual
 345 dependencies using multi-commodity flows, but this approach does not allow for
 346 infrastructure-specific network representations and solution methods.

347 In this paper, the set of physical (resource) dependencies between systems in
 348 S is taken to form a *directed, acyclic graph* (“DAG”) \mathcal{G} that can be ordered with a
 349 topological sort (see [64]). In contrast, geospatial dependencies are not restricted
 350 in such a fashion.

351 3.7. *Critical Flow Centrality*

The **Critical Flow Centrality** (“CFC”) measure reflects the degree to which a given component facilitates the flow of resources to critical locations. Recall that the **flow** in network G (given assignment A) is the aggregate of all flows reaching the demand nodes:

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

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

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

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

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

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

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

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

356 For each time step t , a flow solution $F(t)$ is generated represented in a **sec-**
 357 **ondary graph** G' . This is an adjacency-list representation of the stochastic tran-
 358 sition matrix; every vertex v in G' maintains an *outgoing edge list* in which each
 359 edge is labeled with the probability that a unit of flow travels down that edge.

360 Each edge e and non-demand node v in G' have a *data structure* (i.e., *map*) that
 361 tracks the set of demand nodes reachable from them. Each entry in a map contains
 362 a tuple $(d, P(d|c_{map}))$ giving the probability that a unit of flow passing through
 363 reaches demand node d from the map’s parent component c_{map} . The collection of
 364 all such maps contains the information required to compute Equation 4.

365 The algorithm proceeds by performing a reverse DFS on G' for each demand
 366 node $d \in D$, computing the probability that each edge or non-demand node sends
 367 flow to d . A given node or edge may be visited multiple times in the course of the
 368 search, requiring care to avoid pushing superfluous probability.

```

Function ComputeProbabilities( $G'$ )
  Data:  $G'$ , a graph with components  $(V, E)$  and absorbing nodes
            $D \subseteq V$ .
  foreach  $d \in D$  do
    | ReverseSearch( $G'$ ,  $d$ )
  end
Function ReverseSearch( $G'$ ,  $d$ )
  Data:  $G'$ , as above.
  Data:  $d$ , an absorbing node.
  Var excess[] // array of numbers  $\in [0, 1]$  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: Probability Calculation.

369 Helper variable *excess* is a lookup table containing probability values for each
 370 node. The *IncrementOrAddProbability*() function updates the estimate of $P(d|c)$
 371 stored in the map of component c . The lookup table and variable *amt* are used to
 372 avoid problems with overlapping paths.

373 On typical infrastructure networks, **Algorithm 1** has time and space com-
 374 plexity of $O(|V|^2)$. Each node/link stores $|D|$ entries, one for each demand node
 375 in the network, leading to $O((|V| + |E|)|D|)$ in storage space. The time required to
 376 perform the search for a given demand node is $O(|V| + |E|)$, yielding a total time
 377 of $O((|V| + |E|)|D|)$ for the entire graph. Note that it assumes that the probability
 378 graph induced from the network flow is acyclic. However, the infrastructure net-
 379 works typically have $|V| \approx |E|$ and $|D| \approx \frac{1}{2}|V|$, yielding time and space complexity
 380 bounds of $O(|V|^2)$.

381 The running time of the entire method is thus dominated by the flow generation
 382 step, which is typically more expensive than $O(|V|^2)$. The current paper used the
 383 Edmonds-Karp algorithm (see [64]) for simplicity, which is $O(|V|^2|E|)$ on general
 384 graphs and $O(|V|^3)$ on infrastructure networks. Lastly, note that flows involving
 385 cycles must be handled with other techniques, such as simulation.

386 3.8. An Algorithm for Interdependent Critical Flow Centrality

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

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

Function *ComputeInterdependentCFC*(\mathcal{G})

Data: \mathcal{G} , a graph with nodes $V_{\mathcal{G}} = S = \{S_1, S_2, \dots, S_k\}$ representing
 individual infrastructure systems, and edges $E_{\mathcal{G}}$ formed from
 physical dependencies between elements of $V_{\mathcal{G}}$.

ConvertNetworkRepresentation(\mathcal{G})

Var list \leftarrow *TopologicalSort*($V_{\mathcal{G}}$)

Var t \leftarrow 0

foreach $S_i \in$ list **do**

 | *ComputeSingleSystemCFC*(S_i)

end

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

396

397 **3.8.1. Converting Network Representations**

398 As a pre-processing step, conversion of network representations is performed
 399 to transform each individual network S_i into a format compatible with maximum
 400 flow algorithms.

- 401 1. Nodes with demands are connected to a *supersink* node (see [64, 65]).
- 402 2. Source nodes are connected to a *supersource* node.
- 403 3. Nodes in network S_1 that require resources from network S_2 are represented
 404 in S_2 by corresponding demand nodes.

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

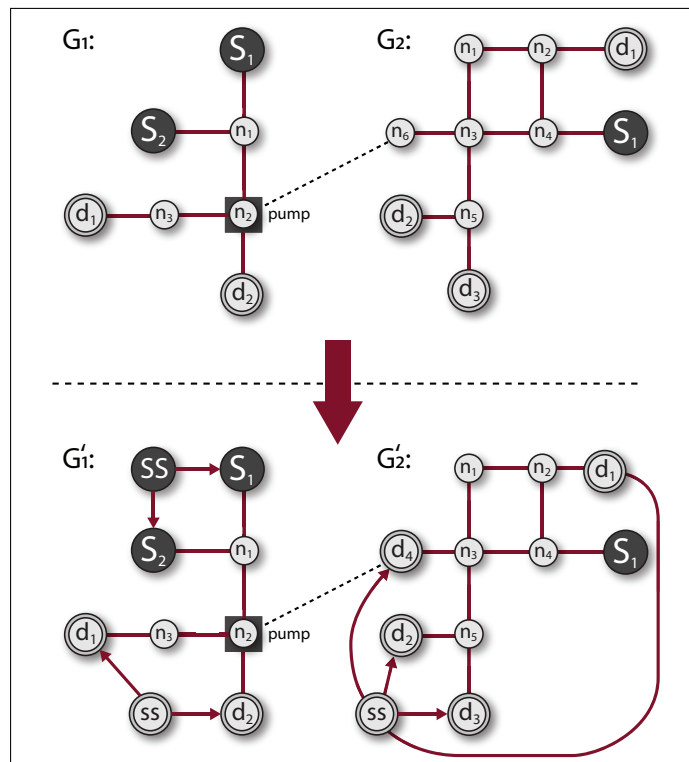


Fig. 3. Two independent infrastructure systems S_1 and S_2 , transformed into flow networks suitable for the Edmonds-Karp algorithm. Supersource ('SS') and supersink nodes ('ss') are added in the usual manner.

409 **3.8.2. Computing CFC Values for a Sub-system**

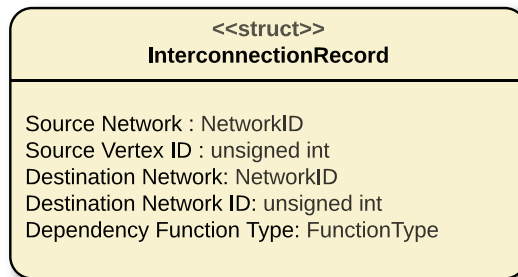
410 Computation of the CFC for sub-system S_i proceeds in two stages: (1) flow
 411 values and criticality values are propagated from other layers S_h ($h < i$) according
 412 to dependencies, and; (2) the CFC for S_i is computed using the technique discussed
 413 in **Section 3.7**. If layer S_i supplies layer S_h with resources (e.g., it is an electricity
 414 network that supplies power to water pumps), then resource demands for S_h appear
 415 in S_i 's network as sinks with appropriate demands. Topological ordering ensures
 416 that S_h 's criticality and flow values have been computed before S_i 's. **Algorithm 3**
 417 provides an overview of single layer CFC computation.

Function *ComputeSingleSystemCFC(S_i)*

PropagateValues(S_i)
ComputeMaxFlow(S_i)
ComputeCFC(S_i)

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

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



420 An interconnection record $IR(v_1, v_2)$ (where $v_1 \in S_h, v_2 \in S_i$) indicates a
 421 physical (resource) dependency between systems S_h and S_i . Demand and criticality
 422 values for v_1 must be propagated to v_2 before the maximum flow and CFC can be
 423 computed for S_i .

424 **Algorithm 4** gives an overview of this process. Criticality values are copied
 425 directly, but the amount of resource that must be provided by v_2 to v_1 is determined
 426 by a function (e.g., the demand induced at v_2 is half of the flow at v_1).

```

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

```

Algorithm 4: Propagation of resource demands.

427 **Figure 4** shows a water system and electricity system that are interlinked in
 428 two locations: pumps near the source of the water system are fed by electricity
 429 nodes labelled *A* and *B*. A flow solution was first computed for the water system,
 430 yielding flows of 6063 litres and 5973 liters at the pumps. The induced demand
 431 at nodes *A* and *B* of the electricity system are half of the flow – namely, 3031 and
 432 2986 units.

433 Note also that edges and vertices with no flow are shown in black. The existence
 434 of such elements is an artifact of the Edmonds-Karp algorithm [65, 64] used in
 435 this simple instantiation, and one that would be corrected by using domain-specific
 436 methods (e.g., hydraulic simulation [55]).

437 **Figure 5** shows the CFC values for the same interdependent infrastructure
 438 system under the same flow solution. Criticality levels (ranging from 0 to 1) are
 439 shown in white font for the buildings. (Lot criticality is fixed at 0.02, and elided
 440 for brevity).

441 Thanks to the propagation of both flow and criticality values from one network
 442 to the next, the criticality of the water pumps is appropriately represented in the
 443 criticality ratings of the electricity system. The electricity nodes *A* and *B* have
 444 inherited criticality values of 0.32 and 0.61 from the corresponding pump vertices
 445 in the water system; they require flow of 3031 and 2986 units, which the reader
 446 can verify by inspection are half of the flow values at the water pump.

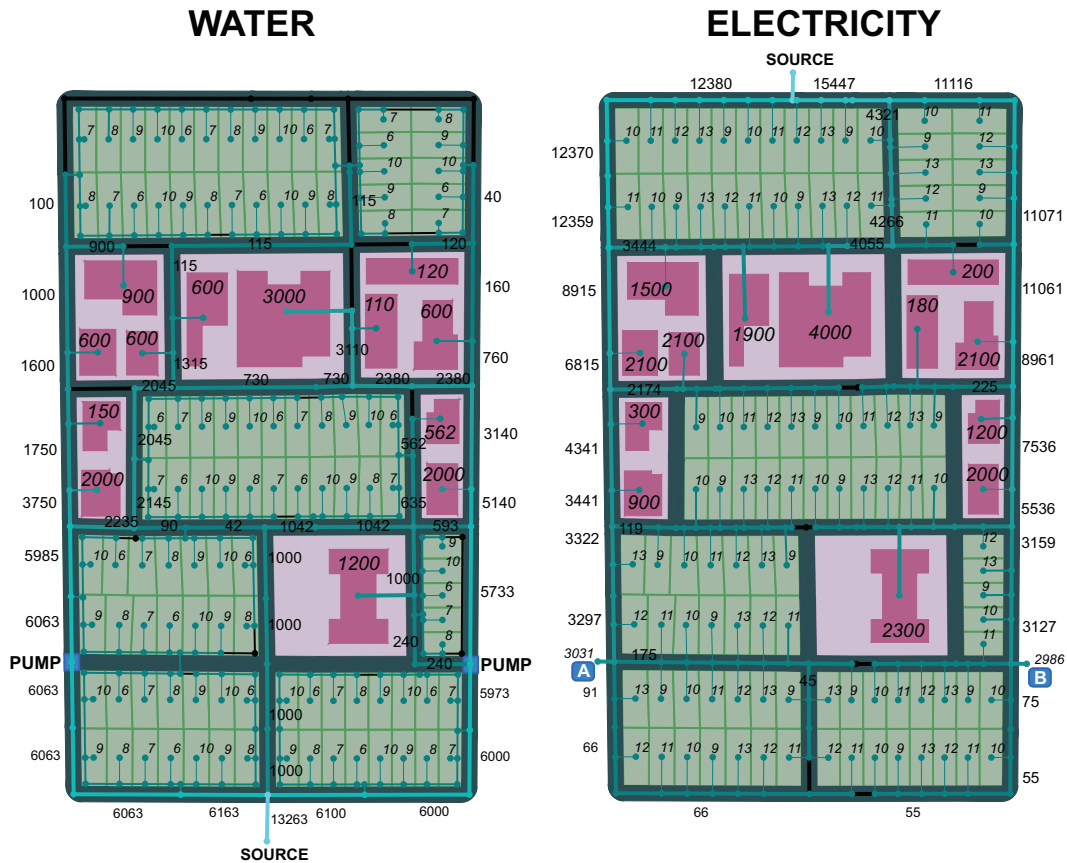


Fig. 4. Interdependent flows. Demand values are in italics while flow values are in regular font. Pumps in the water network are supplied with electricity by nodes A and B. Pumps require electricity proportional to half of their water flow. Black edges/vertices have zero flow.

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

462 series are assumed to give average hourly demands over a 24-hour day. However,
 463 the method is general, and other scenarios could be supported, such as long-term
 464 (i.e., decadal) investigation of urban growth and its effect on capacity.

465 Time series data is assigned to buildings according to type (e.g., secondary
 466 school, restaurant), while lots are assigned time series randomly drawn from a
 467 library of typical residential demand curves. For simplicity, criticality ratings
 468 and vertex/edge capacities are assumed to be static, although they could easily be
 469 represented with their own time series.

470 Empirical data for different types of buildings in summer was obtained from
 471 several sources (e.g., water consumption data was sourced from the California
 472 Public Utilities Commission [66], electricity data from Ontario Power Generation).
 Examples of water demand curves appear in **Figure 6** below:

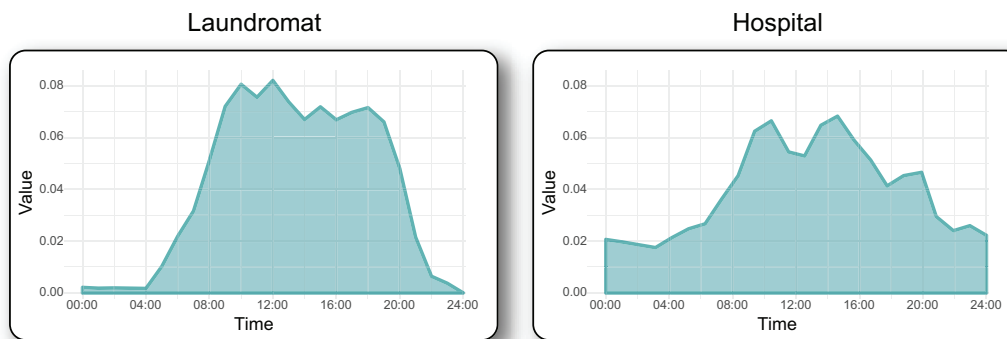


Fig. 6. Hourly time series showing water demands from a laundromat and hospital over an average day. The time series have been normalized to create a probability distribution. For use in the CFC method, these distributions are scaled by average water usage per day.

473
 474 CFC values are computed for each time step $t \in [1, T]$ by loading the relevant
 475 time series data for t and executing **Algorithm 2**. An overview of the process is
 476 provided in **Algorithm 5**. Upon termination of this procedure, each node and edge
 477 in the interdependent system has a set of CFC values — one for each time step —
 478 that can be used in statistical analysis.

479 **Figure 7** shows a graph of CFC values for the water network’s edges over the
 480 full 24-hour cycle:

481 The edge with a constant criticality rating of 1 is the lone edge incident to the
 482 source/reservoir. In general, the edges with significant criticality values tend to
 483 remain critical throughout the 24-hour cycle, with interesting behaviour happening
 484 during the middle of the day. Low criticality nodes become more critical during

Function *ComputeCriticaltyForTimeSeries*(\mathcal{G})

```

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

```

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

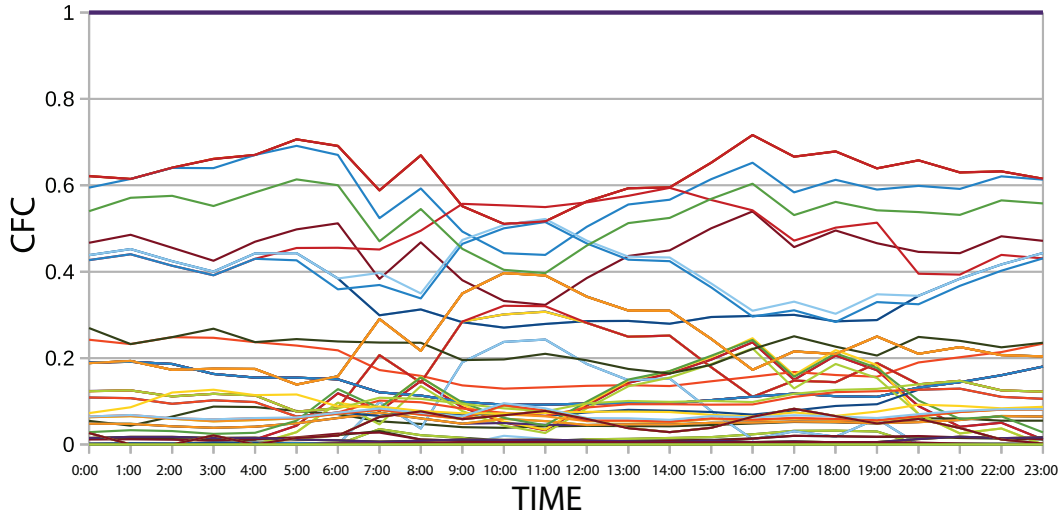


Fig. 7. CFC values for each edge in the water network.

485 mid-day, when significant water demand begins to push capacity constraints.

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

To recap, Algorithm 5 results in a set of CFC values $CFC_t(c)$, where t is a timestep and c is a component. For instance, the output for the water system edges can be represented as a matrix CFC_{water}^e in which rows are timesteps and columns

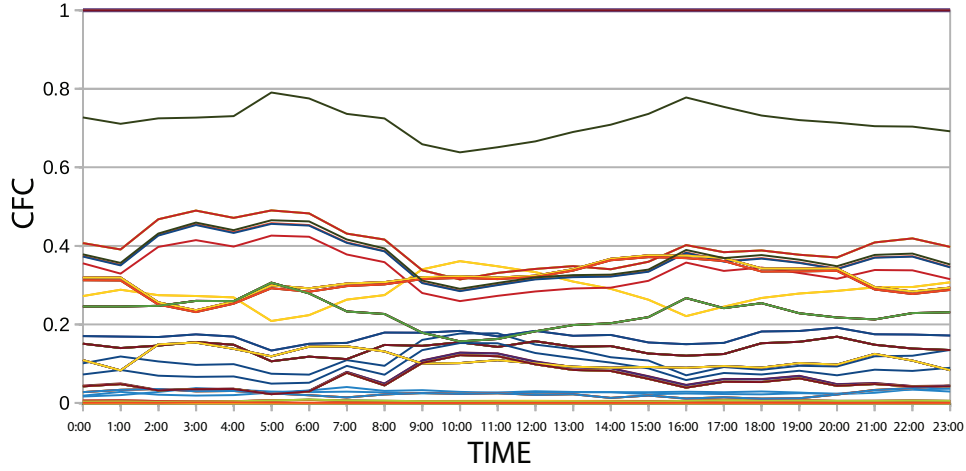


Fig. 8. CFC values for each edge in the electricity network.

are edges:

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

493

494 One major issue not addressed by classical works on network centrality (e.g., [25])
 495 is the choice of ranking method for component measures taken at different times.
 496 The most intuitive approach to ranking the components is to take the *sample mean*
 497 of each column and to subsequently rank columns in descending order. This would
 498 be an appropriate strategy if each row of the matrix was a sample from the space of
 499 assignments (i.e., in a Monte Carlo approach) at a given time t . However, the rows
 500 in the matrix are assessments of the system at different points in time. The use of
 501 descriptive statistical measures (e.g., average, variance) elides system dynamics.
 502 The same is true of various other methods (e.g., spectral analysis, information
 503 theory) that might be employed to analyze the matrix.

504

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

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

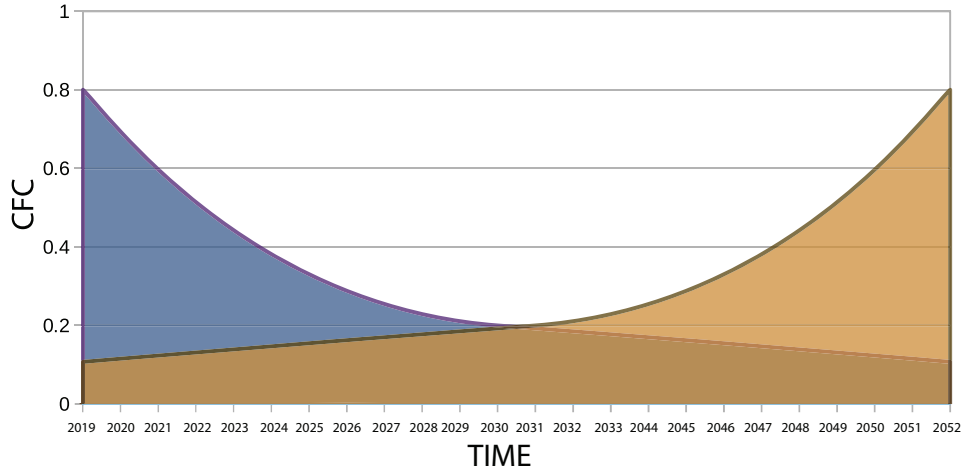


Fig. 9. Two components with similar integrals but different long term behavior.

514 Since the time series in this paper represent average demands in a *daily cycle*,
 515 components are ranked according to the *integral* of their CFC curve. Taking the
 516 water network edges as an example, a cubic spline is defined on the set of sample
 517 points $\{(CFC_1(e), CFC_2(e), \dots, CFC_k(e))\}$ corresponding to edge $e \in E$.

518 An integral is calculated from the cubic spline (as shown in **Figure 10**) and
 519 normalized by the maximum possible area $MAX_CFC \cdot (k - 1) = 1 \cdot 23 = 23$
 520 (recall that CFC values are already normalized, so that the maximum CFC at any
 521 time step is 1). The result is then assigned to the edge e as its *global CFC value*
 522 $CFC_G(e)$ for the entire time series. The set of all water network edges E is then
 523 ranked by sorting the edges according to their CFC_G values.

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

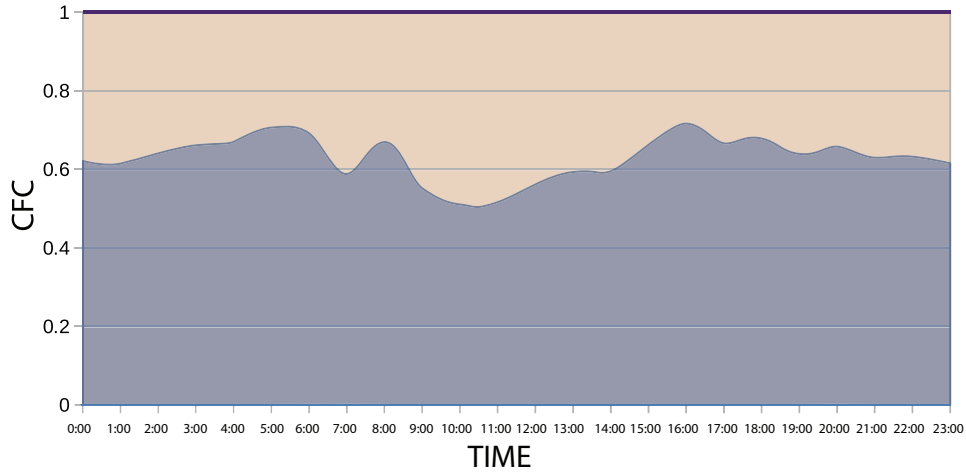


Fig. 10. Computing the global CFC value for a given component e . A cubic spline is overlaid on the CFC values for e . The integral of the spline (blue) is computed and normalized by the total area.

528 5. Reliability

529 This section demonstrates that the CFC measure may be combined with stan-
 530 dard approaches to network reliability — namely, (1) edge reliability measures,
 531 and; (2) ‘leave one out’ failure analysis.

532 5.1. Edge Reliability

An arbitrary network model can be augmented by adding a **reliability function** $r : E \rightarrow [0, 1]$ that assigns edges $e \in E$ a **reliability rating** $r(e) \in [0, 1]$ [67]. One can combine this approach with CFC measures by creating a composite measure that estimates the joint reliability and criticality of a component. For instance, the (normalized) **Unreliable Critical Flow** (“UCF”) for an edge $e \in V$ is:

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

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

537 The computation of the UCF measure can be accomplished with a slight
 538 modification to the algorithm for the CFC. Instead of a static value $r(e)$, the
 539 reliability rating for a network component e can also be represented as a time

540 series $R_e = \{r_{e1}, r_{e2}, \dots, r_{ek}\}$. This allows the modeler to represent different
 541 processes (e.g., decreasing reliability of components over long time periods).

542 The UCF measures are computed for each timestep t using the CFC values
 543 and reliability ratings at t . The end result is a matrix in which entry (i, j) gives
 544 the UCF values for each edge e_j at timestep i . As in the case of the CFC, a cubic
 545 spline is overlaid on the values for each edge, creating an unreliability curve. After
 546 computing the integral and dividing it by the maximum possible area, the *global*
 547 *UCF value* for edge e is computed.

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

554 5.2. 'Leave One Out' Failure Analysis

555 CFC measures can also be used with a common form of reliability analysis
 556 in which components are deliberately failed or degraded (e.g., by reducing their
 557 capacity) in order to assess the effects on the system. A component e may have a
 558 high CFC value under a given assignment, but it may be the case that if e suffers
 559 a (partial) failure there are other routes (i.e., *fallbacks*) through which flow may
 560 travel in order to satisfy critical demand. **Figure 11** illustrates this situation:

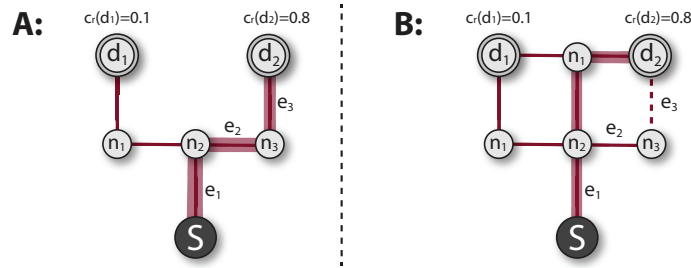


Fig. 11. Two networks with different behaviour in edge failure scenarios. Network A carries most of its critical flow through the path $\{e_1, e_2, e_3\}$. In case of edge failure, no alternative paths are available. Network B has a fallback route in case edge e_3 fails.

561 This form of failure analysis provides an indication of whether there are fallback
 562 routes that can supply critical flow in the event that a component e fails. If the
 563 failure of e consistently results in reduced critical flow across the entire network,

564 one can assume that e is even more critical than the CFC measure alone might
 565 suggest.

Algorithm 6 shows a high-level view of a procedure in which capacities of edges in a single infrastructure system are degraded one-at-a-time. For each time $t < T$, appropriate demands and criticality values are loaded into the graph. Then each edge $e \in E$ is considered in order, degrading its capacity and performing the CFC computation on the altered network. The critical flow is then used to create a *loss measure* that indicates the amount of critical flow that is lost when edge e is degraded. The **failure loss** $FL_t(e)$ for edge $e \in E$ at time t is:

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

566 where (recalling Section 3.4) V_D is the set of demand nodes in G , $\delta(d, t)$ is the
 567 demand at time t from demand node d , $f_A(d, t)$ is the actual flow to d at time t , and
 568 $c_r(d, t) \in [0, 1]$ is the criticality rating for d at t . Failure loss values range from 0
 569 (no effect on resource delivery) to 1 (absolute disruption of resource delivery).

Function *PerformEdgeFailureAnalysis*(S)

```

foreach  $t \in [1, T]$  do
  | LoadDemands( $S, t$ )
  | foreach  $e \in E$  do
  |   | Var originalCapacity  $\leftarrow$  e.capacity
  |   | e.capacity  $\leftarrow$  Degrade(e.capacity)
  |   | ComputeSingleSystemCFC( $S$ )
  |   | ComputeFailureLoss( $S$ )
  |   | e.capacity  $\leftarrow$  originalCapacity
  | end
end

```

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

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

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

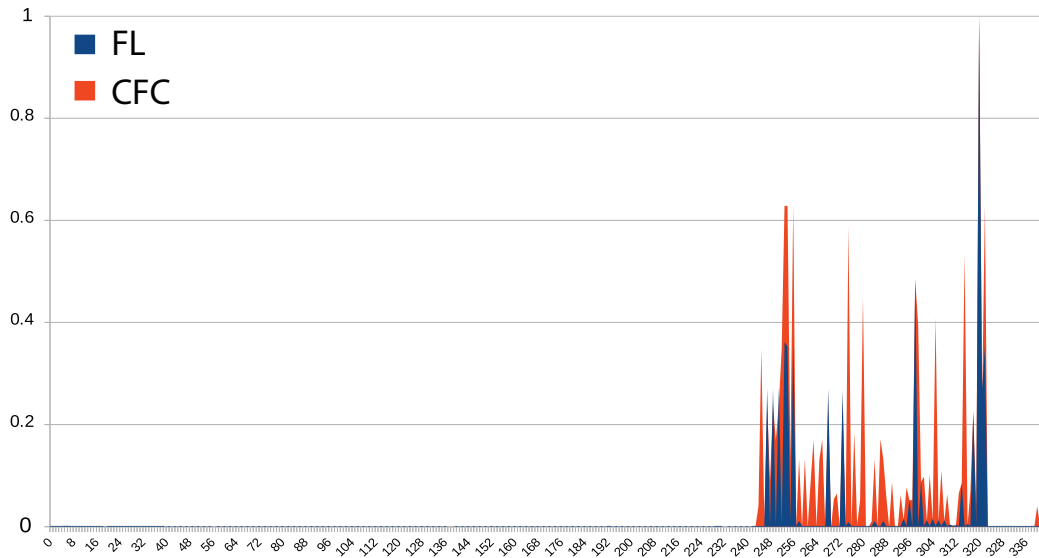


Fig. 12. Averaged *failure loss* (FL) and averaged *critical flow centrality* (CFC) on the edges of the water network in **Figure 4**, computed over a 24-hour period. The majority of edges (e.g., those that feed individual lots) have negligible FL and CFC values.

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

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

592 5.3. Reliability Integration Limitations and Assumptions

593 The edge failure analysis presented above was subject to several simplifying
594 assumptions. First, geospatial dependencies were not included in the analysis for
595 reasons of brevity. Second, the failure loss analysis was performed on a single
596 network instead of a set of interdependent networks. Third, the reliability measures
597 could also incorporate component capacity, in order to capture the intuition that a
598 component nearing its maximum load is likely to be less reliable.

599 Fourth, the definition of the FL metric uses the aggregate of all demands at
600 the network's demand nodes as the normalizing factor. This is appropriate for a
601 network where all demands are satisfied in the baseline state, but it will overestimate
602 losses in networks which exhibit unsatisfied demand. For the scenario utilized in
603 this paper, however, this assumption is reasonable.

604 This method outlined in this work does not assume that the methods used to
605 model each layer are commensurate. That is, the electricity layer may be modeled
606 with one set of domain-specific techniques, while the water layer may be modeled
607 with another. All that is required is for each layer to provide a means of flow
608 computation and a basic network topology. This design decision, while useful
609 from a software engineering perspective, precludes the use of standard approaches
610 to modeling cascading failures.

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

619 6. Conclusion

620 This paper demonstrated how component importance measures based on the
621 notion of critical flow may be applied to interdependent, urban infrastructure
622 systems. The motivation for the work was to provide urban planners and municipal
623 engineers with a method of reasoning about the impacts of interventions on the
624 flow of resources to critical locations. The main theme was that network analysis
625 techniques could be combined with criticality and reliability metrics in order to
626 produce composite methods that provide useful information to stakeholders.

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

633 In the simple variant described in the paper, network flows and ‘critical flow
634 centrality’ (“CFC”) measures were computed using a discrete approach. More
635 sophisticated variants are possible, including the use of domain-specific simulation
636 techniques. For simplicity, the paper assumed that the subsystem dependencies
637 form a directed acyclic graph.

638 The method was demonstrated by use of a simple, district-scale model of a
639 city that contained electricity and water networks. Empirical data was used to
640 estimate resource consumption for different types of buildings, yielding a set of
641 demand curves that represent consumption in a 24-hour cycle. This decision
642 simplified the analysis, and allowed the use of integrals to compute a global CFC
643 value for the entire cycle. For the study of trends in infrastructure systems over
644 time, the integral-based aggregation would need to be supplanted by trend-based,
645 multi-variable time series analysis.

646 Despite the simplifying assumption, the simple method presented in the paper
647 satisfied the goals outlined in **Section 3.1**. First, the computation of CFC metrics
648 for an interdependent system can be computed efficiently. For a model $\mathcal{S} =$
649 $\{S_1, S_2, \dots, S_k\}$ consisting of k subsystems, computation of CFC metrics for \mathcal{S}
650 on typical infrastructure networks is $O(kV^2)$, where V is the average number of
651 nodes in the subsystems. This compares favorably with other centrality measures,
652 which can be $O(V^3)$ or greater.

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

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

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

670 The instantiation of the CFC computation presented in both the current and
671 previous papers are suitable for medium/long time horizons. The main culprit is the
672 use of integer-valued representations for demands and capacity constraints. This
673 decision, which was made in order to simplify the problem and avoid numerical
674 instability, means that short term dynamics are difficult to represent. This precludes
675 forms of analysis in which the rates of change (e.g., of flow) on system components
676 may be analyzed. The use of floating point representations and domain-specific
677 flow computation methods (e.g., simulation) will avoid this restriction.

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

684 **7. Acknowledgements**

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

690 **References**

- 691 [1] T. G. Lewis, Critical Infrastructure Protection in Homeland Security: De-
692 fending a Networked Nation, Wiley-Interscience, 2006.
- 693 [2] R. M. Clark, S. Hakim, A. Ostfeld, Handbook of Water and Wastewater
694 Systems Protection, Springer, 2011.
- 695 [3] J. Lopez, R. Setola, S. D. Wolthusen, Critical Infrastructure Protection:
696 Information Infrastructure Models, Analysis, and Defense, Springer, 2012.

- 697 [4] A. Birolini, *Reliability Engineering Theory and Practice*, 5 ed., Springer,
698 Berlin Heidelberg, 2007.
- 699 [5] N. R. Council, *Drinking Water Distribution Systems: Assessing and Reduc-*
700 *ing Risks*, Technical Report, Committee on Public Water Supply Distribution
701 *Systems: Assessing and Reducing Risks*, 2006.
- 702 [6] D. Meijer, M. van Bijnen, J. Langeveld, H. Korving, J. Post, F. Clemens,
703 *Identifying critical elements in sewer networks using graph-theory*, *Water*
704 *10* (2018).
- 705 [7] A. A. Chowdhury, D. O. Koval, *Power Distribution System Reliability: Prac-*
706 *tical Methods and Applications*, Wiley, Hoboken, New Jersey, 2009.
- 707 [8] M. G. Resende, P. M. Pardalos, *Handbook of Optimization in Telecommu-*
708 *nications*, Springer, New York, NY, 2006.
- 709 [9] M. A. P. Taylor, *Vulnerability Analysis for Transportation Networks*, Elsevier,
710 Amsterdam, Netherlands, 2017.
- 711 [10] M. T. Thai, P. M. Pardalos, *Handbook of Optimization in Complex Networks:*
712 *Theory and Applications*, volume 57, Springer, 2012.
- 713 [11] K. Wolter, A. Avritzer, M. Vieira, *Resilience assessment and evaluation of*
714 *computing systems*, Springer, 2012.
- 715 [12] R. S. Wilkov, *Analysis and Design of Reliable Computer Networks*, *IEEE*
716 *Transactions on Communications* 20 (1972) 660–678.
- 717 [13] I. Gertsbakh, Y. Shpungin, *Network Reliability and Resilience*, Springer,
718 2011.
- 719 [14] S. K. Chaturvedi, *Network Reliability: Measures and Evaluation*, Scrivener
720 Publishing, Hoboken, New Jersey, 2016.
- 721 [15] B. Sudakov, V. Vu, *Local resilience of graphs*, *Random Structures and*
722 *Algorithms* 33 (2008) 409–433.
- 723 [16] M. Krivelevich, C. Lee, B. Sudakov, *Resilient pancyclicity of random and*
724 *pseudo-random graphs*, *SIAM Journal on Discrete Mathematics* 24 (2009)
725 1–17.

- 726 [17] L. Dall'Asta, A. Barrat, M. Barthélemy, A. Vespignani, Vulnerability of
727 weighted networks, *Journal of Statistical Mechanics: Theory and Experiment*
728 (2006) P04006.
- 729 [18] S. Dunn, S. M. Wilkinson, Identifying Critical Components in Infrastructure
730 Networks Using Network Topology, *Journal of Infrastructure Systems* 19
731 (2012) 157–165.
- 732 [19] C. D. Nicholson, K. Barker, J. E. Ramirez-Marquez, Flow-based vulnerabil-
733 ity measures for network component importance: Experimentation with pre-
734 paredness planning, *Reliability Engineering and System Safety* 145 (2016)
735 62–73.
- 736 [20] Y. Almoghathawi, K. Barker, Component importance measures for interde-
737 pendent infrastructure network resilience, *Computers and Industrial Engi-
738 neering* 133 (2019) 153–164.
- 739 [21] K. Stephenson, M. Zelen, Rethinking centrality: Methods and examples,
740 *Social Networks* 11 (1989) 1–37.
- 741 [22] V. Latora, V. Nicosia, G. Russo, *Complex Networks Principles , Methods
742 and Applications*, Cambridge University Press, Cambridge (UK), 2017.
- 743 [23] M. Curado, L. Tortosa, J. F. Vicent, G. Yeghikyan, Analysis and comparison
744 of centrality measures applied to urban networks with data, *Journal of
745 Computational Science* forthcoming (2020). URL: [https://doi.org/10.
746 1016/j.jns.2019.116544](https://doi.org/10.1016/j.jns.2019.116544).
- 747 [24] J. P. Scott, *Social Network Analysis: A Handbook*, SAGE Publications,
748 London, UK, 2000.
- 749 [25] L. C. Freeman, S. P. Borgatti, D. R. White, Centrality in valued graphs: A
750 measure of betweenness based on network flow, *Social Networks* 13 (1991)
751 141–154.
- 752 [26] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, S. Havlin, Catastrophic
753 cascade of failures in interdependent networks, *Nature* 464 (2010) 1025–
754 1028.
- 755 [27] A. Vespignani, Complex networks: The fragility of interdependency, *Nature*
756 464 (2010) 984–985.

- 757 [28] E. Zio, G. Sansavini, Modeling failure cascades in critical infrastructures with
758 physically-characterized components and interdependencies, in: ESREL
759 2010 Annual Conference, 2010, pp. 652–661.
- 760 [29] R. Zimmerman, C. E. Restrepo, The next step : quantifying infrastructure
761 interdependencies to improve security, *Int. J. Critical Infrastructures* 2 (2006)
762 215–230.
- 763 [30] W. Kröger, C. Nan, Addressing Interdependencies of Complex Technical
764 Networks, in: G. D’Agostino, A. Scala (Eds.), *Networks of Networks: The
765 Last Frontier of Complexity*, Springer, 2014, pp. 279–310.
- 766 [31] M. Papa, *Critical Infrastructure Protection II*, Springer, 2008.
- 767 [32] A. Garas, *Interconnected networks*, Springer, 2016.
- 768 [33] F. Petit, D. Verner, J. Phillips, L. P. Lewis, Critical Infrastructure Protec-
769 tion and Resilience—Integrating Interdependencies, in: A. J. Masys (Ed.),
770 *Security by Design*, Springer, 2018, pp. 193–219.
- 771 [34] M. Ouyang, Review on modeling and simulation of interdependent critical
772 infrastructure systems, *Reliability Engineering and System Safety* 121 (2014)
773 43–60.
- 774 [35] G. Bianconi, *Multilayer Networks: Structure and Function*, Oxford Univer-
775 sity Press, 2019.
- 776 [36] S. Boccaletti, G. Bianconi, R. Criado, C. I. Genio, The structure and dynamics
777 of multilayer networks, *Physics Reports* 544 (2014) 1–122.
- 778 [37] Per Hokstad, I. B. Utne, J. Vatn, *Risk and Interdependencies in Critical
779 Infrastructures: A Guideline for Analysis*, Springer, 2012.
- 780 [38] G. D’Agostino, A. Scala, *Networks of Networks: The Last Frontier of Com-
781 plexity*, Springer, 2014.
- 782 [39] H. Amini, K. G. Boroojeni, S. S. Iyengar, P. M. Pardalos, F. Blaabjerg, A. M.
783 Madni, *Sustainable Interdependent Networks: From Theory to Application*,
784 Springer, 2018.

- 785 [40] M. H. Amini, K. G. Boroojeni, S. S. Iyengar, P. M. Pardalos, F. Blaabjerg,
786 A. M. Madni, *Sustainable Interdependent Networks II: From Smart Power
787 Grids to Intelligent Transportation Networks*, Springer, 2019.
- 788 [41] S. M. Rinaldi, J. P. Peerenboom, T. K. Kelly, Identifying, understanding, and
789 analyzing critical infrastructure interdependencies, *IEEE Control Systems
790 Magazine* 21 (2001) 11–25.
- 791 [42] A. Nieuwenhuijs, E. Luijff, M. Klaver, Modeling Dependencies in Criti-
792 cal Infrastructures, in: M. Papa, S. Sheno (Eds.), *Critical Infrastructure
793 Protection II*, Springer, 2008, pp. 205–213.
- 794 [43] E. E. Lee, J. E. Mitchell, W. A. Wallace, Restoration of services in interdepen-
795 dent infrastructure systems: A network flows approach, *IEEE Transactions
796 on Systems, Man and Cybernetics Part C: Applications and Reviews* 37
797 (2007) 1303–1317.
- 798 [44] J. P. Peerenboom, R. E. Fisher, Analyzing cross-sector interdependencies,
799 in: *Proceedings of the Annual Hawaii International Conference on System
800 Sciences*, 2007, pp. 1–9.
- 801 [45] G. E. Apostolakis, D. M. Lemon, A screening methodology for the identi-
802 fication and ranking of infrastructure vulnerabilities due to terrorism, *Risk
803 Analysis* 25 (2005) 361–376.
- 804 [46] L. Duenas-Osorio, J. Craig, B. Goodno, A. Bostrom, Interdependent response
805 of networked systems, *Journal of Infrastructure Systems* 13 (2007) 185–194.
- 806 [47] V. Latora, M. Marchiori, Efficient Behavior of Small-World Networks, *Phys-
807 ical Review Letters* 87 (2001) 3–6.
- 808 [48] M. Newman, *Networks: An Introduction*, 2 ed., Oxford University Press,
809 2018.
- 810 [49] G. Galvan, J. Agarwal, Vulnerability analysis of interdependent infrastruc-
811 ture systems, in: T. Haukaas (Ed.), *Proceedings of the 12th International
812 Conference on Applications of Statistics and Probability in Civil Engineering
813 (ICASP12): Vancouver, Canada, July 12-15 University of British Columbia.,
814 2015.*

- 815 [50] N. K. Svendsen, S. D. Wolthusen, Connectivity models of interdependency in
816 mixed-type critical infrastructure networks, *Information Security Technical*
817 *Report 12 (2007)* 44–55.
- 818 [51] N. K. Svendsen, S. D. Wolthusen, Analysis and statistical properties of
819 critical infrastructure interdependency multiframe models, *Proceedings of the*
820 *2007 IEEE Workshop on Information Assurance, IAW (2007)* 247–254.
- 821 [52] N. K. Svendsen, S. D. Wolthusen, Graph models of critical infrastructure
822 interdependencies, *Lecture Notes in Computer Science 4543 LNCS (2007)*
823 208–211.
- 824 [53] N. K. Svendsen, S. D. Wolthusen, An analysis of cyclical interdependencies
825 in critical infrastructures, *Lecture Notes in Computer Science 5141 LNCS*
826 *(2008)* 25–36.
- 827 [54] J. Williams, Identifying sensitive components in infrastructure networks via
828 critical flows, *engrXiv (2019)*. URL: engrxiv.org/hy2bx. doi:10.31224/
829 osf.io/hy2bx.
- 830 [55] P. Novak, V. Guinot, A. Jeffrey, D. E. Reeve, *Hydraulic Modelling – an*
831 *Introduction: Principles, Methods and Applications*, Spon Press, 2010.
- 832 [56] P. Stride, Super sewer: an introduction to the Thames Tideway tunnel
833 project in London, *Proceedings of the Institution of Civil Engineers - Civil*
834 *Engineering 169 (2016)* 51–51.
- 835 [57] P. Stride, *The Thames Tideway Tunnel: Preventing Another Great Stink*, The
836 History Press, 2019.
- 837 [58] M. Amin, Toward Secure and Resilient Interdependent Infrastructures, *Jour-*
838 *nal of Infrastructure Systems 8 (2007)* 67–75.
- 839 [59] R. F. Austin, D. P. DiSera, T. J. Brooks, *GIS for critical infrastructure pro-*
840 *tection*, CRC Press, 2016.
- 841 [60] M. Mair, R. Sitzenfrei, M. Moderl, W. Rauch, Identifying multi-utility net-
842 work similarities, in: *World Environmental and Water Resources Congress*
843 *2012: Crossing Boundaries, 2012*, pp. 3147–3153.

- 844 [61] M. Mair, J. Zischg, W. Rauch, R. Sitzenfrei, Where to find water pipes
845 and sewers? - on the correlation of infrastructure networks in the urban
846 environment, *Water* 9 (2017).
- 847 [62] R. H. Shumway, D. S. Stoffer, *Time Series Analysis and Its Applications*,
848 Springer, 2017.
- 849 [63] R. Setola, S. Bologna, E. Casalicchio, V. Masucci, An Integrated Approach
850 for Simulating Interdependencies, in: M. Papa, S. Shenoj (Eds.), *Critical*
851 *Infrastructure Protection II*, Springer, 2008, pp. 229–242.
- 852 [64] T. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algo-*
853 *rithms*, 3 ed., MIT Press, Cambridge, MA, 2009.
- 854 [65] R. Ahuja, T. Magnanti, J. Orlin, *Network Flows: Theory, Algorithms, and*
855 *Applications*, Prentice Hall, Upper Saddle River, NJ, 1993.
- 856 [66] Aquacraft, *Embedded Energy in Water Studies, Study 3: End-use Water*
857 *Demand Profiles*, Technical Report, California Public Utilities Commission,
858 Energy Division, 2011.
- 859 [67] E. Zio, From complexity science to reliability efficiency: a new way of
860 looking at complex network systems and critical infrastructures, *International*
861 *Journal of Critical Infrastructures* 3 (2007) 488.