Time-based Critical Infrastructure Dependency Analysis for Large-Scale and Cross-Sectoral Failures

George Stergiopoulos[a], Panayiotis Kotzanikolaou[b], Marianthi Theocharidou[c,*], Georgia Lykou[a], Dimitris Gritzalis[a]

[a] *Information Security & Critical Infrastructure Protection Laboratory, Dept. of Informatics, Athens University of Economics & Business, 76 Patission Ave., GR-10434, Athens, Greece*
[b] *Dept. of Informatics, University of Piraeus, 85 Karaoli & Dimitriou, GR-18534, Piraeus, Greece*
[c] *European Commission, Joint Research Center (JRC), Institute for the Protection and the Security of the Citizen (IPSC), Security Technology Assessment Unit, via E. Fermi 2749, Ispra, I-21027, Italy*

Abstract

Dependency analysis of Critical Infrastructures is a computationally intensive problem when dealing with large-scale, cross-sectoral, cascading *and* common-cause failures. The problem intensifies even more when attempting a *dynamic* time-based dependency analysis. In this paper we extend our previous graph-based, risk analysis methodology to dynamically assess the evolution of cascading failures over time. We employ different growth models to capture slow, linear and fast evolving effects, but instead of using static projections, the evolution of each dependency is "objectified" by a fuzzy control system that also considers the effect of near dependencies. To achieve this, the impact (and, eventually, risk) of each dependency is quantified on a time axis, into a form of many-valued logic. In addition, we extend the methodology to analyze major failures triggered by *concurrent* common-cause cascading events. We develop CIDA, a Critical Infrastructure Dependency Analysis tool, which implements the extended risk-based methodology. CIDA aims to support decision makers to *proactively* analyze dynamic and complex dependency risk paths in two ways:

*Corresponding author
Email addresses: geostergiop@aueb.gr (George Stergiopoulos), pkotzani@unipi.gr (Panayiotis Kotzanikolaou), marianthi.theocharidou@jrc.ec.europa.eu (Marianthi Theocharidou), lykoug@aueb.gr (Georgia Lykou), dgrit@aueb.gr (Dimitris Gritzalis)

Preprint submitted to International Journal of Critical Infrastructure Protection November 9, 2015

(a) to identify potentially underestimated low risk dependencies and reclassify them to a higher risk category before they are actually realized and (b) to simulate the effectiveness of alternative mitigation controls with different reaction time. Thus CIDA can be used to evaluate alternative defence strategies for complex, large-scale and multi-sectoral dependency scenarios and assess their resilience in a cost-effective way.

Keywords: Critical Infrastructure Protection, cascading failures, dependency risk graph, time analysis, resilience.

1. Introduction

Most Critical Infrastructures (CIs) can be modeled as cyber-physical systems whose cyber components control their underlying physical components. CIs are inherently complex systems since they integrate heterogeneous platforms, proprietary systems, protocols and open communication networks. In addition, CIs are usually interconnected and interdependent with other CIs which may belong to different sectors (such as energy, ICT or transportation). According to [1] CIs may have *physical*, *informational* or *logical* dependencies between them. Thus, a failure[1] in one infrastructure may affect the operation of other CIs due to their dependencies. In the case of *geographical* dependency, seemingly independent CIs may be affected by a threat due to their physical proximity. The protection from such types of dependency failures is an active and recent area of research, as shown by the numerous projects on the topic, e.g. DIESIS [3, 4], I2Sim [5], CIPRNet [6]. Dependency modeling, simulation and analysis (MS&A) has been studied extensively, using various approaches. A recent publication [7] overviews existing approaches and categorizes them in the following broad categories: (a) empirical, (b) agent-based, (c) system dynamics based, (d) economic theory based, (e) network based approaches, and others.

[1] Failures in CI operation are meant in a broad sense including accidental failure, natural disasters and deliberate cyber threats [2].

Disruptions or outages in CIs are usually categorized as cascading, escalating, or common-cause [1].

- A *cascading failure* is defined as a failure in which a disruption in an infrastructure A affects one or more components in another infrastructure, say B, which in turn leads to the partial or total unavailability of B.
- An *escalating failure* is defined as a failure in which disruption in one infrastructure exacerbates an independent disruption of another infrastructure, usually in the form of increasing the severity or the time needed for recovering from the second failure.
- A *common-cause failure* occurs when two or more infrastructure networks are disrupted at the same time: components within each network fail because of some common cause. This occurs when two infrastructures are co-located (geographic interdependency) or if the root cause of the failure is widespread (e.g. a natural or a man-made disaster).

1.1. Large-scale and cross-sectoral dependencies

Known examples of large-scale failures caused due to CI dependencies are often related to power transmission networks, such as the 2003 major blackouts in the US, Canada and Europe [8]. The cascading process of a failure has been studied and modeled in the past [9, 10, 11, 12, 13, 14]. Although there is a lack of available statistical data for such failures, recent efforts (*e.g.* [15]), have produced failure statistics based on empirical data reported to the media. One of the key findings is that large-scale CI cascading dependencies occur more frequently than expected. However, the effects do not often cascade deeply, *i.e.* nodes that are 4 or 5 hops away in a dependency chain are rarely affected. Another key finding is that even though most reported initiators of cascading effects are CIs belonging to the ICT and the Energy Sector, most of the times the cascading effects are cross-sectoral, *i.e.* CIs of multiple sectors are affected. This seems reasonable since the source infrastructures (ICT and Energy sector

CIs) usually offer vital services to many other CIs of different sectors, thus creating multiple direct (or 1st-order) dependencies.

1.2. Motivation

Recently, several dependency analysis methodologies and tools have been proposed, focusing either on the impact [16], consequences [17] or on the risk derived from CI dependencies [18, 19, 20, 21, 22] and their potential cascading effect.

Existing methodologies and tools are usually *sector-specific*, oriented to power distribution (e.g. [23]) or to water distribution networks (e.g. [24]). These tools are very useful for low-level analysis of small-scale scenarios; for example to identify the critical components within a power transmission network. However, they may fall short when high-level analysis is needed in order to model large-scale, cross-sectoral scenarios. For example, the identification of dependency paths of high economic or societal risk that affect different sectors. In practice, the CI operators perform risk assessments at an organization-level and may not have knowledge (or interest) on threats coming from other dependent CIs. Such knowledge can be acquired, to some extent, by taking part in international table-top exercises [25]. Even though the economic impact of a failure can be assessed by a CI operator (organizational-wise), the overall impact (or risk) of a given CI failure on other dependent CIs is not a tangible number, especially when dealing with multi-order dependencies.

The need for a high-level multi-sectoral risk assessment has been recognized by international bodies and policies [26]. A high-level risk analysis allows the identification of complex cascade or common-cause risk paths and the comparison of alternative mitigation strategies. Note that a multi-layer risk assessment is not an alternative to organization-wide risk assessments, since these are prerequisite input for a multi-risk analysis. Such analysis would require modeling and analyzing hundreds or even thousands of CIs. If a time-based analysis is to be performed, then complexity is even higher. Unfortunately the computation of the cumulative security risks and the identification of the critical points

of failure is NP-complete and thus suboptimal, yet usable dependency analysis tools have to be developed.

1.3. Contribution

In this paper, we extend our recent work on CI dependency analysis [20, 21, 22]. We design and implement a Critical Infrastructure Dependency Analysis tool (CIDA)[27] which is based on risk analysis and graph modelling. Our extensions are twofold: (a) We introduce *time-based* analysis models to study the evolution of dependency chains during slow, linear or fast evolving cascading failures. Note that each dependency may follow a different time model which is "fine-tuned" for each examined dependency using fuzzy modelling. (b) We model *concurrent* cascading and common-cause failures, in order to effectively analyze major failures.

In order to validate the applicability and efficiency of our approach, we stress-test our tool, using random graphs of up to a thousand nodes (CIs) with randomly selected dependencies. Our tests demonstrate the computational efficiency of CIDA for large-scale scenarios, under reasonable parameters (*i.e.* maximum number of dependencies per node and maximum order of dependencies). As a proof of concept, we also run targeted tests based on data of real cascading and common-cause failures.

CIDA is a *proactive* modelling and security dependency analysis tool for the analysis of *large-scale* and *cross-sectoral* dependency scenarios. It allows risk assessors and CIP decision makers to analyze complex dependency graphs and identify critical dependency chains before an actual threat has occurred. Thus it can reveal underestimated dependency risks that need further attention. Moreover, CIDA may also be used as an efficient tool for the assessment of alternative risk mitigation strategies and therefore help increase CI resilience[2].

[2]Resilience implies the ability to withstand accidental or deliberate threats or incidents [28]. A resilience-oriented approach accepts that failures will occur; thus CIs should implement controls that effectively absorb, adapt to or rapidly recover from disruptive events [29].

2. Building Blocks

This section briefly describes the two main building blocks used in the proposed methodology: (a) the underlying multi-risk dependency analysis methodology for cascading failures and (b) fuzzy modelling, which is used to for the time-based analysis of dependencies.

2.1. Multi-risk dependency analysis methodology

In our previous work, a multi-risk dependency analysis method is presented [20, 21, 22] which makes use of the combined results of existing organization-level risk assessments, already performed by the CI operators, in order to assess the risk of n-order dependencies. To visualize the relationships (dependencies) between CIs, we use directed graphs.

2.1.1. First-order dependency risk

Dependency can be defined as "the one-directional reliance of an asset, system, network, or collection thereof –within or across sectors– on an input, interaction, or other requirement from other sources in order to function properly" [30]. In our approach, dependencies are visualized through graphs $G = (N, E)$, where N is the set of nodes (infrastructures or components) and E is the set of edges (or dependencies). The graph is directional marking dependencies from one CI to another. An edge from node $CI_i \rightarrow CI_j$ denotes a *risk* relation, deriving from the dependency of the infrastructure CI_j from a service provided by infrastructure CI_i. To quantify this relation, we use estimations that quantify the resulting impact ($I_{i,j}$) and the likelihood ($L_{i,j}$) of a disruption being realized. The product of the last two values is defined as the dependency risk $R_{i,j}$ caused to infrastructure CI_j due to its dependency on infrastructure CI_i. The numerical value in each edge refers to the level of the cascade-resulting risk for the receiver due to the dependency. This risk is depicted using a risk scale [1..9] where 9 indicates the most severe risk. All parameters ($L_{i,j}, I_{i,j}$ and $R_{i,j}$) are defined in order to assess the risk of the 1st-order dependencies. The main

input of this method is provided by operators and refers to obvious, upstream dependencies as mentioned above.

Example. The following case describes how our method can be used to model multiple 1st-order dependencies. The cause of the mini telecommunication blackout in Rome (2004) [31] was a flood in a major telecommunication service node in Rome due to a broken pipe that provided water to the air conditioning plant. This caused several circuits to fail including the main power supply. Alternative power generators failed to start due to the presence of water and the batteries of electronic equipment soon followed. In order to perform repairs, the air conditioning plant had to be shut down, which led to overheating of several Telco node devices. This disruption in a main Italian Telecom infrastructure (node A) caused problems and delays in different infrastructures (depicted in Table 1, based on the description of [31]). These include the Fiumicino airport (node B) (closure of check-in, ticketing and baggage services and transfers), the ANSI print agency (node C), post offices (node D), banks (node E) as well as the ACEA power distribution (node F) and the communication network (node G - both between landlines and between landlines and mobiles). In this example, the airport operator (CI_B) has a dependency risk $R_{A,B}$ from the infrastructure CI_A. This risk value refers to the likelihood of the disruption in the telecom node to cascade to the airport ($L_{A,B}$), as well as the societal impact $I_{A,B}$ caused to the airport in the case of such a failure been realized at the source of the 1-st order dependency, *i.e.* CI_A.

2.1.2. Risk of n-order dependencies

Using the 1st-order dependencies as described above, we can proceed in assessing potential n-order cascading risks based on a recursive algorithm [22]. Let $\mathbb{CI} = (CI_1, ...CI_m)$ be the set of all the examined infrastructures. $CI_{Y_0} \rightarrow CI_{Y_1} \rightarrow ... \rightarrow CI_{Y_n}$ denotes a chain of connected infrastructures of length n. The algorithm examines each CI as a potential root of a cascading effect (denoted as CI_{Y_0}) and then computes the Dependency Risk DR exhibited by CI_{Y_n} due to its n-order dependency.

Table 1: Example: Mini Telco blackout 1st-order dependencies

Node: (CI)	**Sector**	**1st-order Effect**
A:(Laurentina-Inviolatella telecom node)	ICT	–
B:(Fiumicino airport)	Transport	Closure of check-in, ticketing & baggage services and transfers
C:(ANSI print agency)	ICT	Data transmission problem
D:(Post offices)	Transport	Delays & service perturbations
E:(Banks)	Finance	Delays & service perturbations
F:(ACEA power distribution)	Energy	Loss of monitoring & control of SCADA
G:(Fixed & mobile networks)	ICT	Partial connectivity loss of land-line & mobile phones

Let $CI_{Y_0} \rightarrow CI_{Y_1} \rightarrow ... \rightarrow CI_{Y_n}$ be a chain of dependencies, $L_{Y_0,...,Y_n}$ be the likelihood of the n-order cascading effect and I_{Y_{n-1},Y_n} be the impact of the $CI_{Y_{n-1}} \rightarrow CI_{Y_n}$ dependency. The cascading risk exhibited by CI_{Y_n} due to the n-order dependency is computed as:

$$R_{Y_0,...,Y_n} = L_{Y_0,...,Y_n} \cdot I_{Y_{n-1},Y_n} \equiv \prod_{i=0}^{n-1} L_{Y_i,Y_{i+1}} \cdot I_{Y_{n-1},Y_n} \tag{1}$$

The *cumulative Dependency Risk* should consider the overall risk exhibited by all CIs within the sub-chains of the n-order dependency. Let $CI_{Y_0} \rightarrow CI_{Y_1} \rightarrow ... \rightarrow CI_{Y_n}$ be a chain of dependencies of length n. The cumulative Dependency Risk, denoted as $DR_{Y_0,Y_1,...,Y_n}$, is defined as the overall risk produced by an n-order dependency, computed by the following equation:

$$DR_{Y_0,...,Y_n} = \sum_{i=1}^{n} R_{Y_0,...,Y_i} \equiv \sum_{i=1}^{n} (\prod_{j=1}^{i} L_{Y_{j-1},Y_j}) \cdot I_{Y_{i-1},Y_i} \tag{2}$$

Equation 2 computes the overall dependency risk as a sum of dependency risks from each affected node in the chain, due to a failure realized in the source node of the dependency chain. Risk computation is based on a risk matrix that combines the likelihood and the incoming impact values of each vertex in the chain. Interested readers are referred to [21] for a detailed analysis of dependency risk estimation.

Moreover, in several cases, the likelihood values are not easy to estimate or

may not be available. This means that while a dependency can be identified between two nodes, the probability for a failure to propagate between two nodes is either unknown or certain (likelihood = 1). In both cases, we can follow a simplified version of equation 2, shown in equation 3, that follows the assumption that if a node fails, the dependent nodes will also fail (likelihood=1). The n-order dependency risk is then calculated as the cumulative impacts on the affected nodes in the dependency chain.

$$DR_{Y_0,\ldots,Y_n} = \sum_{i=1}^{n} R_{Y_0,\ldots,Y_i} \equiv \sum_{i=1}^{n} I_{Y_{i-1},Y_i} \tag{3}$$

2.2. Fuzzy Logic

The multi-risk methodology described above is *static* in time, since equations 1 - 3 are based on the maximum expected impact of each dependency. The values produced by these equations assume that: (a) each dependency chain will always produce its worst case impact (and risk) and (b) all dependencies will exhibit the same impact growth rate. However, in reality neither all CI nodes of a chain will escalate to their maximum consequences nor will they experience the same impact growth rate over the time. For this reason, we will extend our multi-risk methodology to incorporate a dynamic, time-based analysis and to also assess scenarios of partial failures. To model this behaviour, we will incorporate fuzzy set theory. We briefly describe the basic concepts which we will integrate in our time-based analysis described in Section 3.

In contrast to classical set theory, fuzzy logic is an attempt to find approximations of vague groupings, in order to project more objective evaluations of values that are difficult to compute [32]. Fuzzy logic variables provide a truth value that ranges in $[0, 1]$ for a possible outcome. Basically, this truth value corresponds to a membership percentage in a set. Our goal is to approximate the time evolution of a cascading failure using fuzzy approximations of impact evolution for various growth models, similarly to a real failure. For example, an incident might at first have a slow cascading effect to other dependent CIs and as time goes by, failure to restore its operation might lead to catastrophic

effects.

3. A Method for Time-Based Analysis of Cascading and Common Cause Failures

First, we extend the static dependency analysis methodology of [20, 21, 22] to a dynamic time-based analysis model. We use different cascading failure growth models and we apply fuzzy logic, in order to simulate realistic approximations of dynamic cascading failures. In addition, we model combined cascading and common cause failures to simulate the effects of dynamic, large-scale and major disasters.

3.1. Modeling Time Analysis of Cascading Failures

Recall from Section 2.1 that $I_{i,j}, L_{i,j}$ denote the impact (in a Likert scale) and likelihood (%) of a failure experienced in the dependency $CI_i \rightarrow CI_j$. These values are derived from assessments performed at an organizational level by the CI operator. As in most static-based risk assessment methodologies, the impact value $I_{i,j}$ refers to the maximum expected impact (*worst-case* scenario approach) regardless of the time it will take for maximum impact to fully realize after a failure. To model the dynamic time-based analysis, we use the following steps:

1. *Model Definition*: Define different failure growth rates.
2. *Setup*: Using the growth rates, pre-compute all possible expected time-based impact values.
3. *Calculate fuzzy time-based impact values*: For a given Dependency Risk Graph, use the pre-computed expected time-based impact values as input to the fuzzy model, to output the fuzzy approximation of the time-based impact for each dependency.
4. *Assess time-related dependency risks*: For each dependency chain, output the time-based Cumulative Dependency Risk using the fuzzy time-based impact values.

These steps are described in detail in the subsections that follow.

3.1.1. Model Definition

Let $T_{i,j}$ denote the time period that the dependency $CI_i \rightarrow CI_j$ will exhibit its maximum expected impact $I_{i,j}$ and let $G_{i,j}$ denote the expected growth of the dependency failure, *e.g. slow*, *linear* or *fast* evolution of the expected consequences, after the failure.

The values of $T_{i,j}$ and $G_{i,j}$ are provided by the risk assessors of each CI along with the values $I_{i,j}$ and $L_{i,j}$. Finally, let t denote an examined time period after a failure. In the rest of this Section and if there is no ambiguity, we will omit the dependency indices for simplicity reasons, *i.e.* we will use I, T, G, instead of $I_{i,j}, T_{i,j}, G_{i,j}$.

The definition sets for all the above values are Likert scales defined as:

- $I \in [1..9]$, 1 is the lowest and 9 the highest impact.
- $T, t \in [1..10]$. More specifically, we adopt a granular time scale, which uses the following unavailability time periods: $1 = 15$min, $2 = 1$hour, $3 = 3$h, $4 = 12$h, $5 = 24$h, $6 = 48$h, $7 = 1$week, $8 = 2$w, $9 = 4$w and $10 =$ more (> 4w).
- $G \in [1..3]$, where 1 represents a slow, 2 a linear and 3 a fast growth (evolution) of a failure experienced in the examined dependency.

Definition 1. *Let $CI_i \rightarrow CI_j$ be the examined dependency with maximum expected impact I, experienced at time period T after a failure and let G be the growth evolution of the failure. The expected* time related *impact of the dependency, experienced at time t is defined as:*

$$I(t) = \begin{cases} I^{(t/T)}, & \text{if } G = 1 \text{ (slow evolution)} \\ I \cdot (\frac{t}{T}), & \text{if } G = 2 \text{ (linear evolution)} \\ I \cdot \log_T t, & \text{if } G = 3 \text{ (fast evolution)} \end{cases} \quad (4)$$

Obviously, $I(t) = I$ for $t \geq T$ if no mitigation controls are taken at the nodes. Using equation 4, each dependency chain can be modeled with the most

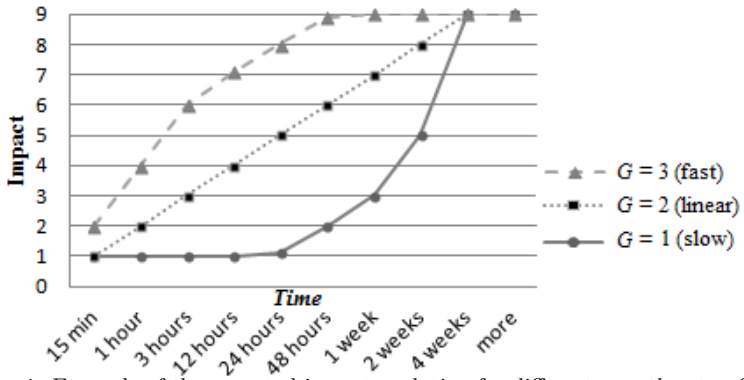

Figure 1: Example of the expected impact evolution for different growth rates, for $I = 9$ and $T = 4$weeks

appropriate evolution growth and a different model is used for each value G. Fast cascading effects are considered to escalate logarithmically (fast short term growth that stabilizes to the maximum expected impact). Linear cascading effects are considered to escalate following a typical linear equation. Slow cascading effects follow an exponential growth rate which starts with low initial values and escalates on the last steps of the time scale. Figure 1 presents a comparison of the different growth rates, for the same I and T.

Example: In a Dependency Risk Graph, the dependencies of CIs connected with a nuclear power facility are modeled with very high impact ($I = 9$) experienced at a relative short time period ($T = 3$h) with a fast evolution ($G = 3$). In the same graph, the edges starting from another typical energy provider are modeled with impact 5 experienced at 48h after the failure and with a linear evolution ($G = 2$). Our model will be able to project the evolution of the dependency risk paths for all time periods, using all different models.

3.1.2. Setup

The growth rates modeled by equation 4 are used to pre-compute all possible values for $I(t)$, for all possible combinations of I, T, G (see Algorithm 1).

Algorithm 1 Calculation of all possible $I(t)$ values

procedure CALCULATEIVALUES($\mathbb{T}, \mathbb{I}, \mathbb{G}$)
  ▷ *Inputs*:
    Time Scale (in min): $\mathbb{T} = \{15, 60, 180, 720, 1440, 2880, 10080, 20160, 40320, 60480\}$
    Scale of Impact: $\mathbb{I} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
    Growth Scales: $\mathbb{G} = \{\text{slow, linear, fast}\}$
  ▷ *Result*:
    All $I(t)$, $\forall G \in \mathbb{G},\ I \in \mathbb{I}$ and $t,\ T \in \mathbb{T}$

  for e_0 in set $\mathbb{G}$ **do**
    Set e_0 as G

    for e_1 in set $\mathbb{T}$ **do**
      Set e_1 as T

      for e_2 in set $\mathbb{T}$ **do**
        Set e_2 as t
        ▷ */* For $t > T$ always output maximum impact */*

        if $t > T$ **then**
          $I(t) = I$

        else if $t \leq T$ **then**

          for I in set $\mathbb{I}$ **do**

            if G is fast **then**
              $I(t) = I \cdot \log_T t$

            else if G is linear **then**
              $I(t) = I \cdot (\frac{t}{T})$

            else if G is slow **then**
              $I(t) = I^{(\frac{t}{T})}$

            end if

          end for

        end if

      end for

    end for

  end for
end procedure

The output of this algorithm will be used as input to a fuzzy logic ranking system, to assess more realistic evolutions of potential failures. The fuzzy logic

classification system uses the following membership sets:

1. The *impact set*: partitions the [1..9] impact scale to groups Very Low, Low, Medium, High and Very High as:
   $\{\texttt{VL}=\{1\},\texttt{L}=\{2,3\},\texttt{M}=\{4,5\},\texttt{H}=\{6,7\},\texttt{VH}=\{8,9\}\}$.
2. The *time set*: partitions the [1..10] time scale to groups Early, Medium, Late and Very Late periods as:
   $\{\texttt{E}=\{1,2,3\},\texttt{M}=\{4,5,6\},\texttt{La}=\{7,8\},\texttt{VLa}=\{9,10\}\}$.

In order to support the fuzzy mechanism (described in Section 3.1.3), the output of Algorithm 1 is stored in pre-computed tables[3]. These provide the expected time related impact values for all possible G and T.

Table 2: An example of a pre-computed table describing the expected time related impact values for fast evolving failures ($G=3$) exhibiting their worst impact at $T=12$ hours

	Time Related Impact	Very Low 1	Low 2	Low 3	Medium 4	Medium 5	High 6	High 7	Very High 8	Very High 9
Early	15 minutes	1	1	2	2	2	3	3	3	4
	1 hour	1	2	2	3	3	4	4	5	6
	3 hours	1	2	3	3	4	5	5	6	7
Medium	12 hours	1	2	3	4	5	6	7	8	9
	24 hours	1	2	3	4	5	6	7	8	9
	48 hours	1	2	3	4	5	6	7	8	9
Late	1 week	1	2	3	4	5	6	7	8	9
	2 weeks	1	2	3	4	5	6	7	8	9
V.Late	4 weeks	1	2	3	4	5	6	7	8	9
	> 4 weeks	1	2	3	4	5	6	7	8	9

An example is shown in Table 2, which describes all possible time-based impact values $I_{i,j}(t)$, assuming a fast evolving failure ($G=3$), experiencing the worst-case impact at time $T=12$h, depicted by the yellow horizontal line. Each column's worst-case impact value is appointed to cells in that row ($T=12$h) and obviously to all rows below that one, since in this scenario the worst-case impact has already been realized at that time. The time-based impact values for all the rows above this ($T=15$min up to $T=3$h) are calculated by applying an inverse growth rate on the cell values in the $T=12$h row, using equation 4

[3]All tables are available at the tool's Wiki: `https://github.com/geostergiop/CIDA/wiki/`

for $G=3$. Notice that the impact and time values have been grouped according to the fuzzy impact and time membership sets respectively. For example, in Table 2, the fuzzy impact membership set '`Low`' contains impact values from 1 up to 3. All the values have been produced by applying the fast (logarithmic) growth scale and the time T = 12h as the expected time of occupance of the worst-case impact value. The output of the algorithm is computed during the setup and it is stored in 30 tables: for each of the three growth rates G, we need to pre-compute and store one table for each of the ten possible T values.

3.1.3. Calculating Fuzzy Time-based Impact Values

By using the pre-computed tables with all expected time-based impact values $I(t)$, it is now possible to assess the fuzzy estimation of the time-related impact values, for a given Dependency Risk Graph.

For each dependency, we use the growth rate G and the expected time T of the worst-case impact value I, to select the corresponding table from the database. Then, fuzzy sets for all impact membership sets (Very Low, Low, Medium, High, Very High) are generated, using the corresponding columns of the selected table. These fuzzy sets along with linguistic IF-THEN rules are used to calculate the fuzzy value of the expected time-based impact value. Rules are usually expressed in the form: "IF variable IS property THEN *action*". All IF-THEN rules are invoked, using the constructed membership sets as linguistic variables to determine an output result, the *fuzzy time-based impact value*, denoted as:

$$Fuzzy(\ I, G, T, t\ ) = \mathcal{I}(t) \tag{5}$$

The calculation of $\mathcal{I}(t)$ is computed as follows. Initially a processing stage invokes the appropriate IF-THEN rules[3] and generates a result for each rule. Then these results are combined to output a set of truth values. Each IF-THEN result is, essentially, a membership function and truth value controlling the output set, *i.e.* the linguistic variables impact and time. The final stage in order to get a single quantitative value from the fuzzy output of the membership

values is "defuzzification", in which all IF-THEN output results are combined to give a single fuzzy time-based impact value for each time point in the time scale. We use the RightMostMembership defuzzification technique [33], which outputs the most-right (*i.e.* highest) impact values, since this is coherent with risk-based standards that tend to favor worst-case scenarios.

The output fuzzy time-base impact values $\mathcal{I}(t)$ are considered more objective approximations of the expected impact at a given time, since instead of simply using the appropriate pre-computed expected time-based impact $I(t)$, the fuzzy values also consider their neighbouring values. Thus they tend to be closer to real-world scenarios. In short, each dependency has its own expected growth but it will also be affected by the growth of its near dependencies.

Example: An examined dependency has input data $G=3$ and $T=12$h, thus the fuzzy mechanism will select Table 2. The group `Low` of the fuzzy impact set (columns 3 and 4) contains only values $1, 2, 3$. By using the aforementioned mechanism, the fuzzy membership set *Low* has membership percentages defined as: $Low = \{(1, 0.05)\ (2, 0.55)\ (3, 0.4)\}$ where the second value is each couple is the the membership percentage of the corresponding impact value. The subset of the rules used to calculate the *Low* output set of the values $\mathcal{I}(t)$ are the following:

```
17: IF Impact IS Low AND Time IS Early THEN Fuzzy_Impact is Very_Low;
18: IF Impact IS Low AND Time IS Medium THEN Fuzzy_Impact is Medium;
19: IF Impact IS Low AND Time IS Late THEN Fuzzy_Impact is High;
20: IF Impact IS Low AND Time IS Very_Late THEN Fuzzy_Impact is High;
```

Let us calculate the fuzzy impact for $I=2$. The fuzzy set mostly characterized by that value is the `Low` set. Let us also assume that the worst-case scenario for this example is at $T=12$h. Based on Table 2, this time belongs to the `Medium` time fuzzy set. Thus, using rule 18, CIDA's setup process will choose to output a `Medium` fuzzy time-based impact value. Last but not least, the tool will now use the aforementioned RightMostMembership defuzzification technique on all these sets to discrete the time-based impact value.

3.1.4. Time-related Multi-order Dependency Risk

The static model described in Section 2.1 can be now extended to provide multiple estimations of the evolutions of the dependency risk, by replacing the static impact value $I_{i,j}$, with the dynamic fuzzy time-related impact values $\mathcal{I}_{i,j}(t)$ described above. These values are used to extend equations 2 and 3 to calculate the n-order dependency $R_{Y_0,\ldots,Y_n}$ and cumulative Risk $DR_{Y_0,\ldots,Y_n}$ of a risk graph for each point on the time-scale as:

$$DR_{Y_0,\ldots,Y_n}(t) = \sum_{i=1}^{n} R_{Y_0,\ldots,Y_i} \equiv \sum_{i=1}^{n} (\prod_{j=1}^{i} L_{Y_{j-1},Y_j}) \cdot \mathcal{I}_{Y_{i-1},Y_i}(t) \tag{6}$$

or, if likelihood assessments are omitted:

$$DR_{Y_0,\ldots,Y_n}(t) = \sum_{i=1}^{n} R_{Y_0,\ldots,Y_i} \equiv \sum_{i=1}^{n} \mathcal{I}_{Y_{i-1},Y_i}(t) \tag{7}$$

Obviously, equations 6 and 7 will produce ten different values, one for each different examined value of t. Examples of test scenarios are given in Section 6.

3.2. Combining cascading and common-cause failure risks

The Dependency Risk assessed by equations (6) or (7), assumes a single initiating event (disruption) at a single CI that results in cascading disruptions. It does not cover common-cause failures which simultaneously affect several, seemingly independent CIs. Such events can cause multiple cascading chains, *i.e.* impact is introduced to multiple nodes in the graph simultaneously. Thus, we extend the model to also capture failures that are at the same time common-cause and cascading failures. A variety of incidents can serve as initiating events of such cases: An accident or a natural disaster, or it can be a human-initiated attack. For example, a common-cause initiating event may concurrently affect CIs –not been identified as directly dependent to each other– due to physical proximity (*e.g.* a flood) or due to social parameters (e.g. a national strike).

Let L_e be the likelihood of an event (threat) e. In the case of natural disasters, the value of L_e can be assessed based on statistics of previous inci-

dents, prognostics and the presence of vulnerabilities, whereas the likelihood of adversarial attacks is more complex. In that case, likelihood is affected by the motivation and the attack skills of the adversary, as well as his perceived impact of the attack. For this reason, the use of expert opinion is commonly applied, coupled with a worst-case approach, to achieve a maximum valuation of risk.

We first use equation 6 (or its simplified impact-only version of equation 7) to evaluate all possible n-order dependency risks. Let $\mathbb{CI}$ be the set of all the examined CIs. The *combined Common-cause Risk*, $CR(CI_{Y_0}, e)$ of all possible chains of cascading events $CI_{Y_0} \rightarrow CI_{Y_1} \rightarrow CI_{Y_n}$, initiated by a common-cause failure event e, for each possible source infrastructure $CI_{Y_0} \in \mathbb{CI}$ can be computed as the sum of all the possible risk chains $DR_{Y_0,\ldots,Y_n}$, $\forall Y_0 \in \mathbb{CI}$, multiplied by the likelihood L_e of each examined event e:

$$CR(\mathbb{CI}, e) = L_e \cdot \sum_{\forall Y_0 \in \mathbb{CI}} DR_{Y_0,\ldots,Y_n}(t) \tag{8}$$

Every CI (node) that is affected by a common-cause event e is examined as a possible root of a dependency chain (as CI_{Y_0}). For each CI_{Y_0}, the cumulative dependency risk DR is computed by applying equation (6) or the simplified version (7). Examples of the combined Common-cause Risk evaluation are provided in section 6.3.

4. Design and Implementation of the CIDA tool

In this section we describe the building blocks used for the design and implementation of the Critical Infrastructure Dependency Analysis (CIDA) tool [27], which extends and implements the dependency analysis methodology described in section 3.

4.1. The Neo4J graph database

We chose a graph database model as the main building block for the development of CIDA. Graph databases are defined as storage systems that provide

index-free adjacency. Graph database technology is an effective tool for modeling data, in comparison with relational databases and querying languages, in cases where the relationship between elements is the driving force for the design of the data model [34, 35]. In a graph database, every element (or node) only needs to know the nodes with which it is connected (*i.e.* its edges). This allows graph database systems to utilize graph theory in order to efficiently examine the connections and degree of nodes' connectivity. In addition, the edge utility allows a graph database to find results in associative data sets. Graph databases can scale more naturally to large data sets or to datasets with frequently changing or on-the-fly schemas [34].

We examined various graph databases and we selected the Neo4J [36] framework to implement our dependency analysis modeling tool, due to its adaptability, scalability and efficiency. According to recent empirical studies (e.g. [37, 35, 38]), Neo4J outperforms other systems in load time for thousands of elements, as well as in the time needed to compute the total paths and detect the shortest path. On the other hand, Neo4J shows inferior performance when used in highly volatile network topologies (*i.e.* graphs with frequent changes in nodes and edges), in comparison to other graph model approaches (e.g. DEX, Titan-Cassandra). This however does not affect our model, since CI dependencies are not subject to frequent changes in their connectivity.

Neo4J builds upon the property graph model; nodes may have various labels and each label can serve as an informational entity. The nodes are connected via directed, typed relationships. Both nodes and relationships hold arbitrary properties (key-value pairs). Although there is no rigid schema, the node-labels and the relationship-types can provide to the nodes as much meta-information as necessary for the node attributes required by a specific schema. When importing data into a graph database, the relationships are treated with as much value as the database records themselves [34].

Neo4j deploys a single server instance that can handle a graph of billions of nodes and relationships. If data throughput exceeds a limit, the graph database can be distributed among multiple servers thus providing a scalable configura-

tion with high availability. In addition, Neo4J's listeners are capable to capture useful signals (notices) that objects broadcast in response to possible events, such as a change of a property value or a user interaction. As explained in Section 5, these two graph functions are the most important factors affecting the computation time for the analysis of dependent and interconnected CIs.

During development, the Blueprints technology was also used: a property graph model interface which provides implementations to support the Neo4J graph database.

4.2. Design of the Dependency Graph Analysis Tool

We developed CIDA, a graph-based Critical Infrastructures Dependency Analysis tool that implements the methodology described in section 3, in order to dynamically analyze the risk of CI dependency chains, for cascading and/or common-cause failures. CIDA is able to compute the security risk and/or the impact evolution of the dependencies over the time. CIDA can graphically represent complex graphs of thousands of dependent CIs through a weighted, directed graph. The weight of each connection (edge) between two CIs, is the (maximum) estimated dependency risk value deriving from the dependency between two infrastructures. In order to make the computation of the risk dependency paths more efficient, we need to set a maximum depth limit of the examined dependencies. Based on recent empirical research results [15], cascading effects rarely affect CIs beyond 5th-order dependencies and thus we use this as an upper limit to the order of dependencies which are evaluated.

The tool takes as input (either from a spreadsheet or using a graphical interface) the nodes and the edges of the graph. For each edge $CI_i \rightarrow CI_j$ the following input is required for a static analysis: the estimated likelihood L_{ij} and the (maximum) expected impact I_{ij}. For a dynamic time-based analysis, the expected time T_{ij} of the maximum impact and the expected growth rate G_{ij} for the dependency are also required. For a given input dependency graph, CIDA outputs the following:

1. A table of all existing dependency paths, up to a given maximum dependency order (5-th order by default).
2. For a static analysis (based only on L_{ij}, I_{ij}) and for each dependency path, it computes the Cumulative Dependency Risk using equation 2.
3. For a dynamic analysis (additionally using the time related input T_{ij}, G_{ij}) it computes the expected Cumulative Dependency Risk for various time frames, using the formula of equation 6.
4. The dependency paths can be sorted based on their cumulative Dependency Risk value and can also be presented in a graphical form. If the assessor has set a maximum risk threshold, then CIDA also highlights all the paths exhibiting a value higher than the risk threshold. The paths can be exported to several forms, such as a spreadsheet or an XML file for further analysis.
5. In addition, CIDA graphically highlights the path exhibiting the maximum cumulative Dependency Risk path, using a modified version of the Dijkstra algorithm. The algorithm uses negative weights (risk values) to compute the maximum weighted path.

Based on the assessor's preferences, CIDA can compute both cyclic paths (if feedback effects are to be considered) or acyclic paths (if feedback effects are excluded). In cases where the likelihood values are not available, CIDA can proceed using only the impact values and equations 3 and 7 can be used for a static or dynamic analysis of dependency *impact* paths respectively. Obviously, impact estimations will be higher than risk estimations, as they represent worst-case scenarios that take for granted that if a node fails, then all the following nodes will experience a total failure.

4.3. Modeling CIs & Dependencies

In CIDA, each node may represent a CI or an "autonomous" sub-component of a CI (e.g. a power generation sub-station in the case of a power operator), depending on the required level of analysis. Each node in a graph supports the following attributes:

- *Name:* A unique name for the node.
- *CI operator:* The CI owner, responsible for the proper operation of this node. If the analysis is performed in a unit-level, then one operator may be responsible for several nodes.
- *CI sector:* The sector of the CI operator. For example Energy or ICT sector.
- *CI sub-sector:* The specific sub-sector that the operator of the CI belongs to. For example Electricity production or Telecom operator.
- *Node Location:* (Location latitude and longitude) Used to capture geographical dependencies between CIs and evaluate potential threats concurrently affecting several nodes.
- *maxPath:* A boolean that simple indicates if the node belongs to the maximum risk path.

CIDA supports 17 different CI sectors, including communications, energy, transportation, water systems and all the recognized CI sectors, based on the reports of [28] and [39]. It also allows to model all types of dependencies. While *logical*, *informational* and *physical* dependencies may be defined in Service Level Agreements (SLAs) and, thus, are easier to identify, *geographical* dependencies may be missed. CIDA can automatically identify geographical dependencies, based on the location provided for each node. This enables CIDA to study threat scenarios within a specific geographical region and thus to assess geographical dependencies.

Based on the appropriate formulas described in section 3, CIDA computes the risk for each individual dependency path. Then the risk values are then stored in the Neo4J graph, which is implemented as a weighted, directed graph. It then creates a visualization interface using the JUNG2 graph visualization library, supported by Blueprints and Neo4J.

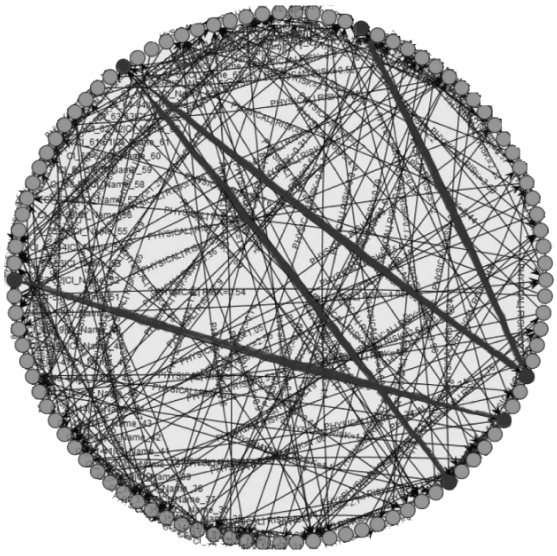

Figure 2: Calculating a dependency risk graph for a test scenario with 100 nodes

An example of a graph output is shown in figure 2. Each node represents a CI and the weight of each edge is the estimated dependency risk value that derives from the dependency between two nodes. The type of the dependency is also depicted. Red edges and nodes indicate the maximum cascading risk path. The complete set of paths and the relative risk values are exported to a spreadsheet, for further analysis. Examples of complete input and output sets are provided in Section 6.

5. Efficiency analysis

We empirically study the efficiency of the CIDA dependency analysis tool presented in section 4 using granular random scenarios as test cases, containing from 10 up to 1000 nodes. All the tests were performed on an Intel Core i-7 processor @2.7GHz with 4 cores and 16 GB of RAM. For each scenario, we examined two different cases with different degrees of connectivity. In the low connectivity case, each node is randomly connected with at least 1 and at most 3, other nodes. In the high connectivity case, each node is randomly connected with at least 4 and at most 5 other nodes. For each case, we multiplied the

execution time by a factor of 10, to estimate the execution time for dynamic analysis, since for time-based analysis all the computations are repeated for each examined time frame.

In real-world scenarios the majority of nodes are expected to have a connectivity degree ≤ 3, while only a small portion of nodes is expected to experience a high degree of connectivity (*e.g.* major electrical production sub-stations or key Telco node) [15]. Thus it is reasonable to expect that in a real-world scenario the average connectivity degree of all the nodes will be between these values and the expected execution time will be between these setups (for the same number of nodes).

Each test was repeated 10 times and the mean execution time was considered. The execution time includes the calculation of *all* the dependency risk paths up to 5th-order dependencies, and the time required to short the paths according to the execution time and compute the maximum dependency risk path. Again, since the cascading effects rarely affect nodes at a distance greater that 4 hops away from the source of the event, the computation of up to the 5th-order dependencies suffices to cover the majority of the cases [15]. As expected, the computation time of the complete set of existing dependency risk paths increases exponentially with the number of nodes (see figure 3). Also, the connectivity degree seems to significantly affect the execution time. For example, in the scenario of 1000 nodes it takes about 20 min in the case of low connected nodes (1-3 edges per node), while it requires more than 332 min for higher connectivity ($4-5$ edges per node).

Since the maximum path problem is NP-complete, it is obvious that CIDA cannot provide a complete theoretical solution. However, with reasonable limitations in system parameters such as the maximum number of nodes, the degree of node connectivity and the degree of the dependency order, CIDA can efficiently provide useful results for large-scale and cross-sectoral realistic scenarios.

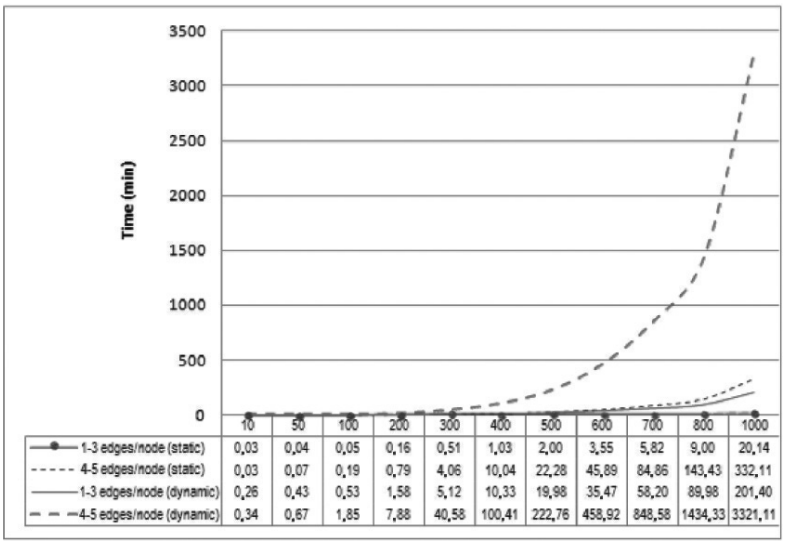

Figure 3: Execution time vs number of nodes

6. Analyzing Real-World Scenarios of Cascading Effects

To demonstrate the applicability of CIDA, we run scenarios based on a known real-world case of cascading effects. Although our tests are based on a real case, the impact, likelihood and time-related input assigned to each dependency do not rely on actual risk assessment results, since such assessments are not publicly available, but they are based on our subjective assumptions for demonstration purposes.

6.1. Case study: A real-world cascading blackout failure

We run a scenario consisting of 9 nodes, based on the well-known electricity blackout scenario of California [1]. The scenario was selected because it is a well-documented case, which exhibits several complex cross-sectoral and multi-order cascading dependencies. The initiating event in this scenario is the failure of an electric power sub-station (node A –see figure 4). This event triggered the following 1st-order dependencies: the disruption to a natural gas production infrastructure (node B), the disruption to the operation of petroleum pipelines

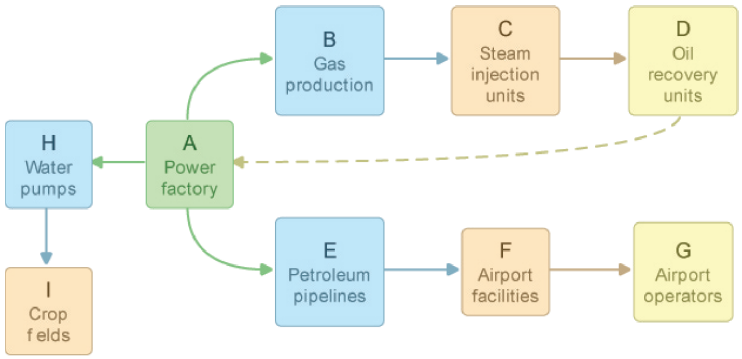

Figure 4: Dependencies of the California blackout scenario

(node E) transporting jet fuel within neighbor states, and problems in the operation of massive water pump units (node H).

As depicted in figure 4, the disruption to gas production (node B) directly impacted gas supplies for steam injection units (node C, – 2nd-order dependency). The steam injection units affected the operation of heavy oil recovery units (node D, – 3rd-order dependency), further exacerbating power problems to node A (feedback loop). Similarly, the disruption of product petroleum pipelines (node E) caused inventories to build up at refineries and draw down at the product terminals (2nd-order dependency), including several major California airports (node F). The reduction of jet fuel stocks at the airports caused several major airline operators (node G) to consider contingency plans (3rd-order dependencies). Finally, the disruption of the water pump units (node H) affected crop irrigation at crop fields (node I, –3rd-order dependency).

For our case study, we used as input the values shown in table 3. For each dependency, we provide a likelihood and a maximum expected impact estimation. We also provide the expected time that the maximum impact will manifest, as well as an estimation for the growth rate of the failure evolution.

Note that the input data of the 1st-order dependencies can be fed to CIDA either through a spreadsheet, as the one shown in table 3, or they can be graphically added/modified (an example is shown in figure 5). Based on the data input, CIDA will compute the complete set of all the dependency risk paths

Table 3: Assumed input values for the examined blackout scenario.

CI_i	CI_j	$L_{i,j}$	$I_{i,j}$	$T_{i,j}$	$G_{i,j}$
A	B	0.9	8	24h	*fast*
B	C	0.65	6	48h	*linear*
C	D	0.7	6	48h	*slow*
D	A	0.15	3	2w	*slow*
A	H	0.8	6	12h	*fast*
H	I	0.65	7	48h	*linear*
A	E	0.9	8	12h	*fast*
E	F	0.7	5	24h	*slow*
F	G	0.25	8	1w	*slow*

in a time-axis, for each dependency chain of order ≤ 5, using the formula of equation 6.

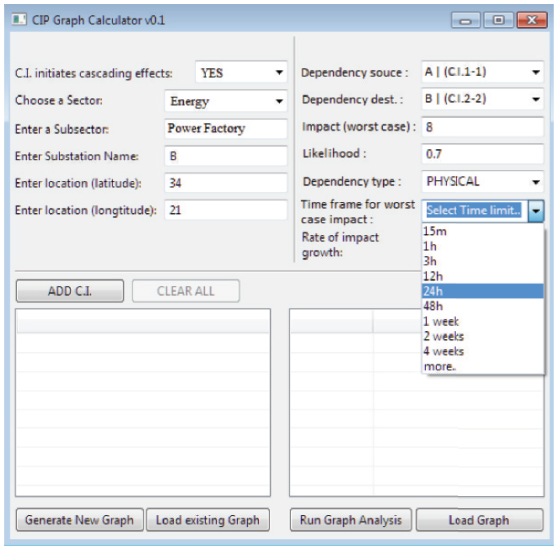

Figure 5: Using the graphical interface to input data for nodes (CIs) and connections (edges)

CIDA will output a graphical representation of the examined Dependency Risk Graph, as shown in figure 6. In this case, the graph of the 9 examined nodes produces 38 chains of order between 2 and 5 and of potential risk value between 1.5 and 13.17. The nodes and edges indicated with red color show the maximum Cumulative Dependency Risk path.

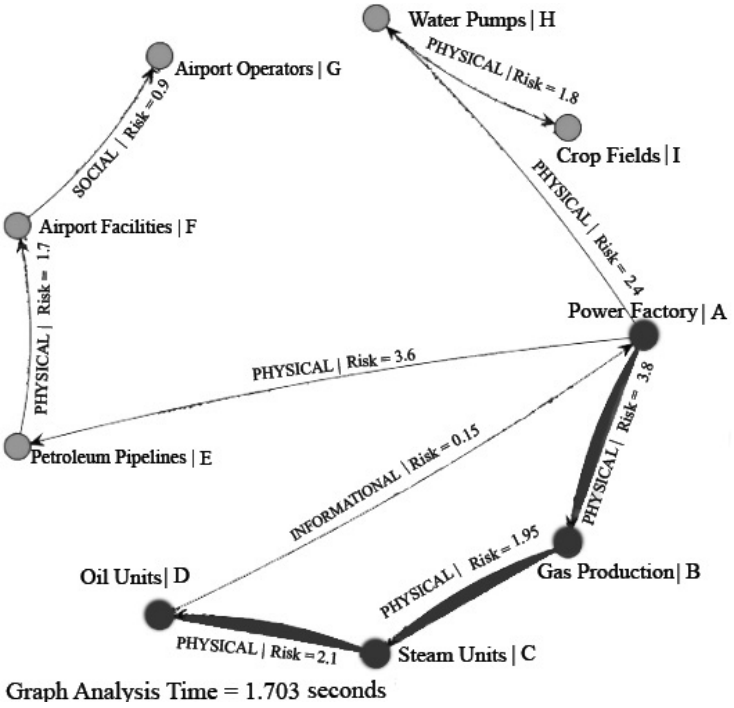

Figure 6: The Dependency Risk Graph of the examined blackout scenario

6.2. A cascading-only dependency failure scenario

After CIDA has evaluated all the Dependency Risk paths, it is now possible for the assessor to efficiently examine several scenarios. One scenario is to analyze and compare all possible cascading effects. CIDA produces a list of all dependency paths, sorted by their Cumulative Dependency Risk value. This may help the assessor to identify all the potential dependency risks that are above a threshold value. For example, figure 7 shows the subset of the dependency risk paths that exhibit a cumulative risk above 5, regardless of the time required to reach this threshold. It also shows the risk values of the maximum risk path for all the examined time periods, as well as the maximum risk level for the rest paths at the time of occurrence.

The threshold parameter is chosen by the assessor and may assist the decision makers in implementing the most effective risk mitigation strategies. From the above figure it is easy to see that the four dependency paths with the highest

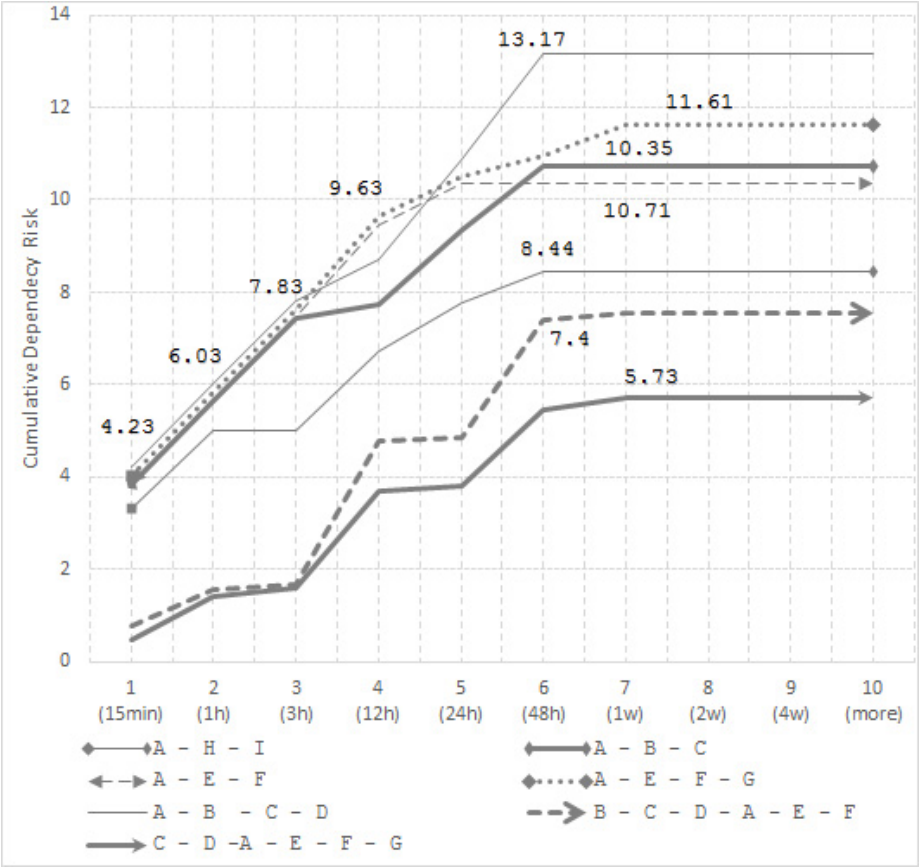

Figure 7: Dependency risk paths with Cumulative Dependency Risk above a risk threshold equal to 5. The assessor may project different paths by using a different threshold

risk value, surpass the threshold only within 1 hour after an initial failure and they all start at node A. Thus a cost-effective mitigation strategy would start by applying mitigation controls at node A with fast response time, which would substantially decrease the overall dependency risk. If mitigation techniques at node A are considered too expensive, an alternative strategy would be to reduce the likelihood of cascade to the most important 2nd-order dependencies of node A (nodes B and E).

A second result of the analysis is that, although path (A-B-C-D) exhibits the highest risk for almost all examined time frames, still we can see from the graph that path (A-E-F-G) is the most critical path around 12 hours after a cascading

failure. This is due to the fact that although both dependencies A-B and A-E have a fast growth, the second dependency is expected to have the fastest convergence to its maximum impact (T_{AE} = 12h). Recall that our methodology can model different T_{ij} and G_{ij} for each dependency.

A third result can be derived by comparing the evolution of sub-paths exhibiting high risk. For example, note that although the path (A-B-C-D) is the highest risk path, its sub-path (A-B-C) already exhibits impact higher than the threshold within 1 hour. Thus it is necessary to implement mitigation controls at the first or second order dependency.

Finally, for the rest of the dependency paths (A-H-I, B-C-D-A-E-F and C-D-A-E-F-G) it is safe to consider mitigation controls with a slower response time, since they have relatively low risk, even for long time frames. Note that the last two paths include the chain (A-E-F) as sub-path thus mitigation controls implemented at (A-E-F) will concurrently reduce the risk of 4 out of the 7 most critical paths.

6.3. Combined common-cause and cascading scenarios

Another scenario is to analyze all the cascading effects that may potentially be triggered by a common-cause failure. For each examined node, CIDA computes the sum of the dependency risks of all the existing distinct paths[4] originating from it. In the examined case, node A is (as expected) by far the most critical node for common-cause failures, having a sum of distinct risk paths equal to 33.22 (paths A-B-C-D, A-E-F-G and A-H-I). In a common cause scenario, at least two nodes are directly affected by the initiating event, which serves as the common cause for the failures. Therefore, for each affected node, we would calculate the sum of distinct risk paths and then these values would be weighted with the likelihood value L_e for the initiating event e (failure or attack) that may be realized at these source nodes. The combined risk (for each possible initiating event e at each directly affected node) is computed using the

[4]In order to avoid repetition of the same risk chain, only distinct paths are considered.

formula of equation 8.

Note that the complete set of the dependency chain risks has already been computed as an output of the CIDA tool. Thus the evaluation of the possible common-cause failures will be based on "ready to use" risk chains. The assessor is able to add initiating events and likelihood values for every node, reflecting the probability that the examined event will cause a failure to the node or not. The examined initiating events e and likelihood values L_e are selected by the assessor based on expert opinions and possible statistical data. Obviously it is reasonable to first examine nodes that exhibit the higher sum of risk values (before they are weighted with L_e).

7. Discussion

Ouyang [7] distinguishes modeling and simulation approaches as (a) empirical, (b) agent based, (c) system dynamics based, (d) economic theory based, (e) network (topology or flow) and (f) others (hierarchical holographic modeling, high level architecture based method, petri-net based method, dynamic control system theory based, Bayesian network based, etc.)[5] Our method can be categorized as a hybrid one, having characteristics of empirical methods (as a risk based approach) and network based methods (as a graph modeling tool).

Empirical methods have been criticized for the lack of the required statistical data needed to assess the likelihood of potential events. While probability data may be difficult to collect for various CIs, efforts have already been made for specific sectors. For example, Carreras et al. [23] demonstrate statistical studies of blackouts which allow the identification of critical power lines (or groups of power lines) for a given network model. This provides a technique for identifying critical clusters of lines that are likely to trigger or propagate cascade effects, due to vulnerabilities of their power lines.

[5] Another taxonomy of dependency models contains six broad categories [40, 14]: (a) aggregate supply and demand tools, (b) dynamic simulations, (c) agent-based models, (d) physics-based models, (e) population mobility models and (f) Leontief input-output models.

Our approach also inherits from network based methods as it combines a method for discovering dependency risk paths with an automated modeling and analysis tool. It allows the dependencies of the connected infrastructures to be depicted in a graph and critical paths to be identified. Such flow-based, network approaches exist in the literature. They either model the flow of products or services between CIs in a uniform model [41, 42, 43] or they combine different sector-based flow models [44, 45].

Most modeling, simulation and analysis (MS&A) tools in the literature are sector-specific. For example, in the water sector, OpenMI [46] addresses standards for federated modeling and simulation for a wide range of technical, organizational and economic aspects related to water (sea, dikes, ground water, water management, and more). Other approaches allow for *integrated* or *federated* simulation, combining models from various sectors. Examples include the DIESIS [3, 4], EPIC [47] and I2Sim [5] approaches.

Regarding the level of analysis, CIDA does not model infrastructures on a component level, and for this reason it has similarities with the empirical Leontief input-output models used for high level multi-sectorial risk assessment [48, 49, 50, 51]. These approaches measure the dependencies among CI sectors by economic relationships. The approach described in [51] considers domain expert opinion and uses fuzzy logic to assess the impact induced by direct and higher-order dependencies between CIs. These models assume that the CI operators (or the expert(s) conducting the assessment) will provide input data regarding the impact values for resource outages of various durations on each CI. Both CIDA and [51] use fuzzy logic. The approach of [51] uses it to minimize the uncertainty and ambiguity associated with the subjective information obtained from domain experts. On the other hand, CIDA combines fuzzy logic with various time growth models. Each dependency may follow a different growth rate and fuzzy logic is used in order to objectify the evolution of each dependency, taking into consideration the state of other near dependencies. This allows CIDA to output results both for various time frames (this is also available in [51]) and for alternative growth rates of the failure. Moreover, CIDA is based

on a dependency risk graph to model dependencies, which are not limited to economic dependencies.

The approaches of [48, 49, 50] use the input-output inoperability model to assess the dependencies between the various sectors of an economy and forecasting the effect on one segment of a change in another one, due to a disruptive event. Our approach is not a purely economical one.Another important variation is that CIDA allows alterative graphs to be created to analyze the dependencies that occur in abnormal operating conditions; in contrast, the input to the approaches of [48, 49, 50] only measure dependency in normal economic operations. Moreover, CIDA can perform time-based analysis, which offers different risk results according to the time frame studied and the rate that the impact evolves in each node. Regarding the input data required for the analysis, the above approaches use aggregated economic data on a sector basis, while CIDA's input consist of risk assessment data provided by the CI operator, *i.e.* per each infrastructure.

Another economic based approach is the NISAC tool N-ABLE, which is a large-scale microeconomic simulation tool that models the complex supply-chain, spatial market dynamics, and CI interdependencies of businesses in the U.S. economy. N-ABLE has been designed in particular to model how U.S. businesses can adapt to and recover from disruptive events [52]. Our tool is not specifically engineered to model the economic impact on a microeconomic level.

The CIP/DSS (Critical Infrastructure Protection/Decision Support System) [53, 54] enables decision makers to determine what consequences might be expected from disruptions to infrastructure, explore the mechanisms behind these consequences, and evaluate mitigation controls for a particular risk. CIP/DSS assesses uncertainties in threats, vulnerabilities, and the consequences of terrorist acts and natural disasters at a national and metropolitan scale. It models interdependencies for all U.S. identified critical infrastructures and key resources and calculates the impact that cascade into these interdependent infrastructures and into the national economy. Our tool could benefit by the CIP/DSS representation of mitigation alternatives, in order to formalize more the selection of

mitigation strategies.

Based on our analysis, CIDA can efficiently compute the risk of all the dependency risk paths, using reasonable limits for the length of the dependency order. However when examining large-scale scenarios of hundreds of nodes, the execution time may not be feasible for *real-time* analysis and response. The tool CIPR/Sim [55] is such a real-time analysis tool, which imports real-time data from numerous existing analysis modules, including RTDS (Real Time Digital Simulator) for electric grid analysis, QualNet for telecommunications analysis, and PC Tide for wind speed and flood surge analysis.

8. Conclusions

CIDA is a modeling and analysis tool focusing on the study of large-scale dependency scenarios for *proactive* analysis. The primary goal of CIDA is to help risk assessors and CIP decision makers to assess dependency risks proactively, before an initiating threat has been realized. By analyzing the complete set of the potential dependency paths, the risk assessors may project all the cascading effects that may potentially be realized and thus flag dependency risks above a threshold that need further attention.

In addition, CIDA can be used to run specific scenarios which may be of particular interest for the risk assessors. While the exhaustive computation of the complete set of dependency risk paths may provide useful information and reveal "hidden" dependency risks, the assessors may also use CIDA to specifically examine particular realistic scenarios. For example the risk assessors may use CIDA to examine "what-if" scenarios that only consider initiating security events affecting one (or some) nodes. Such a scenario may include the study of a major physical disaster, initially affecting all the nodes within a geographical range. This can be easily implemented, since the attributes of each node in CIDA may also include the geographical location coordinates. Thus it is possible to assess scenarios of common-cause failures to targeted nodes and examine cascading effects based on geographical dependencies (in such a

case, the most usual type of the corresponding 1st-order dependencies will be geographical).

CIDA may also be used as an efficient tool for the proactive assessment of risk mitigation controls and therefore, increasing resilience. Based on real input data, it is feasible to examine hundreds of variant scenarios, even past incidents. The risk assessors may efficiently run slight variations of dependency graphs, with different weights or even different dependencies, in order to simulate the implementation of alternative risk mitigation controls. For example, if a particular path has been identified by the exhaustive computation as a path of a dependency risk above the maximum risk threshold, CIDA can be used to project the effect of implementing redundancy security controls to decrease the dependency impact to a targeted edge; or the effect of completely substitute an edge (if security controls that completely remove a dependency were implemented), in order to optimize the topology of the interdependent CIs. Both examples increase the *absorbing* capacity of nodes or of the network of CIs and, thus, increase overall resilience.

CIDA can also be used in order to identify and target key nodes, in order to make them more *resistant* to failures or improve their *restorative* capabilities, and which are the alternative resilience parameters that could be improved. In this way it will be possible to evaluate the benefit of various alternative and/or complementary mitigation controls, and output convincing arguments concerning the expected benefit of possible mitigation strategies. Note that the study of such scenarios can be parallelized by using additional computing resources.

A limitation of our approach is the need for input data of prior risk assessments, performed by the examined CIs. This is an inherent problem of all the empirical risk approaches, since empirical risk-based approaches analyze dependencies based on previous incidents (historical incident or disaster data) coupled with expert opinion in an effort to identify alternative measures to minimize the dependency risk (see for example [18, 19, 16]). It is highly unlikely for a single CI operator to have access to real data of other CIs. Thus the methodology can

only be applied at a higher layer. For example sector coordinators or regulators may collect data concerning a specific sector (such as ICT or energy). National CIP authorities or CERTs may be able to collect such information. Also note that input data can be gradually added in the CIDA database in order to gradually construct large-scale scenarios. Moreover, it is also possible to use such a tool based only on the expected impact of dependent CIs, and ignoring the likelihood parameter, which is generally more difficult to collect [16]. We plan to enrich CIDA [6]in its next version with available statistical data for potential initiating events and their likelihood of occurrence, for key sectors such as Energy and ICT, in order to further assist risk assessors in evaluating combined common-case and cascading failures.

Acknowledgment

M. Theocharidou's work has been performed under the CIPRNet project. This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 312450. The European Commission's support is gratefully acknowledged. Preliminary stages of this work have been supported by the Excellence and Extroversion Programme (Action 2) of Athens University of Economics and Business (AUEB). The authors wish to thank David Ward for his suggestions, which have greatly improved the quality of this paper.

References

[1] S. Rinaldi, J. Peerenboom, T. Kelly, Identifying, understanding, and analyzing critical infrastructure interdependencies, IEEE Control Systems Magazine 21 (6) (2001) 11–25. doi:10.1109/37.969131.

[6]The entire CIDA project (both the stress test implementation along with the full GUI CIDA tool can be found at: https://github.com/geostergiop/CIDA

[2] R. Kozik, M. Choras, Current cyber security threats and challenges in critical infrastructures protection, in: Informatics and Applications (ICIA),2013 Second International Conference on, 2013, pp. 93–97. doi:10.1109/ICoIA.2013.6650236.

[3] A. Usov, C. Beyel, E. Rome, U. Beyer, E. Castorini, P. Palazzari, A. Tofani, The DIESIS approach to semantically interoperable federated critical infrastructure simulation, in: Advances in System Simulation (SIMUL), 2010 Second International Conference on, IEEE, 2010, pp. 121–128.

[4] E. Rome, S. Bologna, E. Gelenbe, E. H. Luiijf, V. Masucci, DIESIS: an interoperable european federated simulation network for critical infrastructures, in: Proceedings of the 2009 SISO European Simulation Interoperability Workshop, Society for Modeling & Simulation International, 2009, pp. 139–146.

[5] J. R. Marti, J. A. Hollman, C. Ventura, J. Jatskevich, Dynamic recovery of critical infrastructures: real-time temporal coordination, International Journal of Critical Infrastructures 4 (1) (2008) 17–31.

[6] Critical infrastructure preparedness and resilience research network (2014).
URL http://www.ciprnet.eu/

[7] M. Ouyang, Review on modeling and simulation of interdependent critical infrastructure systems, Rel. Eng. & Sys. Safety 121 (2014) 43–60.

[8] G. Andersson, P. Donalek, R. Farmer, N. Hatziargyriou, I. Kamwa, P. Kundur, N. Martins, J. Paserba, P. Pourbeik, J. Sanchez-Gasca, R. Schulz, A. Stankovic, C. Taylor, V. Vittal, Causes of the 2003 major grid blackouts in North America and Europe, and recommended means to improve system dynamic performance, IEEE Transactions on Power Systems 20 (4) (2005) 1922–1928. doi:10.1109/TPWRS.2005.857942.

[9] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, S. Havlin, Catastrophic cascade of failures in interdependent networks, Nature 464 (2010) 10251028.

[10] S. Panzieri, R. Setola, Failures propagation in critical interdependent infrastructures, International Journal of Modelling, Identification and Control 3 (1) (2008) 69–78.

[11] A. Vespignani, Complex networks: The fragility of interdependenc, Nature 464 (2010) 984985.

[12] D. Zhou, A. Bashan, Y. Berezin, R. Cohen, S. Havlin, On the dynamics of cascading failures in interdependent networks, arXiv preprint arXiv:1211.2330.

[13] L. Dueñas-Osorio, S. M. Vemuru, Cascading failures in complex infrastructure systems, Structural safety 31 (2) (2009) 157–167.

[14] E. Zio, G. Sansavini, Modeling interdependent network systems for identifying cascade-safe operating margins, Reliability, IEEE Transactions on 60 (1) (2011) 94–101.

[15] M. Van Eeten, A. Nieuwenhuijs, E. Luiijf, M. Klaver, E. Cruz, The state and the threat of cascading failure across critical infrastructures: the implications of empirical evidence from media incident reports, Public Administration 89 (2) (2011) 381–400.

[16] L. Franchina, M. Carbonelli, L. Gratta, M. Crisci, An impact-based approach for the analysis of cascading effects in critical infrastructures, International journal of critical infrastructures 7 (1) (2011) 73–90.

[17] B. Robert, A method for the study of cascading effects within lifeline networks, International journal of critical infrastructures 1 (1) (2004) 86–99.

[18] I. B. Utne, P. Hokstad, J. Vatn, A method for risk modeling of interdependencies in critical infrastructures, Reliability Engineering & System Safety 96 (6) (2011) 671–678.

[19] G. H. Kjølle, I. B. Utne, O. Gjerde, Risk analysis of critical infrastructures emphasizing electricity supply and interdependencies, Reliability Engineering & System Safety 105 (2012) 80–89.

[20] P. Kotzanikolaou, M. Theoharidou, D. Gritzalis, Interdependencies between critical infrastructures: Analyzing the risk of cascading effects, in: S. Bologna, B. M. Hämmerli, D. Gritzalis, S. D. Wolthusen (Eds.), CRITIS, Vol. 6983 of Lecture Notes in Computer Science, Springer, 2011, pp. 104–115.

[21] P. Kotzanikolaou, M. Theoharidou, D. Gritzalis, Assessing n-order dependencies between critical infrastructures, IJCIS 9 (1/2) (2013) 93–110.

[22] P. Kotzanikolaou, M. Theoharidou, D. Gritzalis, Cascading effects of common-cause failures in critical infrastructures, in: J. Butts, S. Shenoi (Eds.), Critical Infrastructure Protection, Vol. 417 of IFIP Advances in Information and Communication Technology, Springer, 2013, pp. 171–182.

[23] B. Carreras, D. Newman, I. Dobson, Determining the vulnerabilities of the power transmission system, in: System Science (HICSS), 2012 45th Hawaii International Conference on, 2012, pp. 2044–2053. doi:10.1109/HICSS.2012.208.

[24] D. Judi, A. Kalyanapu, S. Burian, B. Daniel, T. McPherson, Wide-area flood inundation and infrastructure risk assessment simulation framework, in: Proceedings of the Second IASTED International Conference on Water Resources Management, 2007.

[25] H. Luiijf, D. Stolk, An international tabletop exercise on critical infrastructure protection: the lessons identified, International journal of critical infrastructures 6 (3) (2010) 293–303.

[26] The post 2015 Hyogo Framework for Action: Managing risks to achieve resilience achieve resilience, European Commission, Brussels, 2014, COM(2014) 216 final.

[27] Y. Stergiopoulos, P. Kotzanikolaou, M. Theocharidou, D. Gritzalis, CIDA: Critical Infrastructure Dependency Analysis tool (September 2014).
URL https://github.com/geostergiop/CIDA

[28] Presidential Policy Directive - Critical Infrastructure Security and Resilience (PPD-21), The White House, 2013.

[29] R. Francis, B. Bekera, A metric and frameworks for resilience analysis of engineered and infrastructure systems, Rel. Eng. & Sys. Safety 121 (2014) 90–103.

[30] NIPP 2013: Partnering for Critical Infrastructure Security and Resilience, Dept. of Homeland Security, 2013.

[31] W. Kröger, Emerging risks related to large-scale engineered systems, Tech. rep., International Risk Governance Council (2010).

[32] I. Perfilieva, J. Močkoř, Mathematical principles of fuzzy logic, Springer, 1999.

[33] W. V. Leekwijck, E. E. Kerre, Defuzzification: criteria and classification, Fuzzy sets and systems 108 (2) (1999) 159–178.

[34] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen, D. Wilkins, A comparison of a graph database and a relational database: a data provenance perspective, in: Proceedings of the 48th annual Southeast regional conference, ACM, 2010, p. 42.

[35] B. Shao, H. Wang, Y. Xiao, Managing and mining large graphs: systems and implementations, in: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, ACM, 2012, pp. 589–592.

[36] Neo4J graph database (2014).
URL http://www.neo4j.org/

[37] S. Jouili, V. Vansteenberghe, An empirical comparison of graph databases, in: Social Computing (SocialCom), 2013 International Conference on, 2013, pp. 708–715. doi:10.1109/SocialCom.2013.106.

[38] S. Batra, C. Tyagi, Comparative analysis of relational and graph databases, International Journal of Soft Computing and Engineering (IJSCE) 2 (2) (2012) 509–512.

[39] Green Paper on a european programme for critical infrastructure protection, Commission of the European Communities, 2005, COM(2005) 576 final.

[40] W. Kröger, E. Zio, Vulnerable systems, Springer, 2011.

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

[42] N. K. Svendsen, S. D. Wolthusen, Connectivity models of interdependency in mixed-type critical infrastructure networks, Information Security Technical Report 12 (1) (2007) 44–55.

[43] N. K. Svendsen, S. D. Wolthusen, Analysis and statistical properties of critical infrastructure interdependency multiflow models, in: Information Assurance and Security Workshop, 2007. IAW'07. IEEE SMC, IEEE, 2007, pp. 247–254.

[44] M. Ouyang, L. Dueñas-Osorio, An approach to design interface topologies across interdependent urban infrastructure systems, Reliability Engineering & System Safety 96 (11) (2011) 1462–1473.

[45] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. Porcellinis, R. Setola, Modelling interdependent infrastructures using interacting dynamical models, International Journal of Critical Infrastructures 4 (1) (2008) 63–79.

[46] J. Talsma, B. P. BECKER, Q. Gao, E. RUIJGH, Coupling of multiple channel flow models with openmi, in: 10th International Conference on Hydroinformatics HIC, 2012.

[47] C. Siaterlis, B. Genge, M. Hohenadel, EPIC: A testbed for scientifically rigorous cyber-physical security experimentation, IEEE Transactions on Emerging Topics in Computing 1 (2) (2013) 319–330. doi:10.1109/TETC.2013.2287188.

[48] Y. Y. Haimes, P. Jiang, Leontief-based model of risk in complex interconnected infrastructures, Journal of Infrastructure systems 7 (1) (2001) 1–12.

[49] J. R. Santos, Y. Y. Haimes, Modeling the demand reduction input-output (i-o) inoperability due to terrorism of interconnected infrastructures*, Risk Analysis 24 (6) (2004) 1437–1451.

[50] J. R. Santos, Inoperability input-output modeling of disruptions to interdependent economic systems, Systems Engineering 9 (1) (2006) 20–34.

[51] R. Setola, S. De Porcellinis, M. Sforna, Critical infrastructure dependency assessment using the input–output inoperability model, International Journal of Critical Infrastructure Protection 2 (4) (2009) 170–178.

[52] M. A. Ehlen, A. J. Scholand, Modeling interdependencies between power and economic sectors using the n-able agent-based model, in: Proceedings of the IEEE power engineering society general meeting, 2005, pp. 2842–2846.

[53] B. Bush, L. Dauelsberg, R. LeClaire, D. Powell, S. Deland, M. Samsa, Critical infrastructure protection decision support system (cip/dss) project overview, in: Proceedings of the 23rd international conference of the system dynamics society, 2005, pp. 17–21.

[54] S. H. Conrad, R. J. LeClaire, G. P. O'Reilly, H. Uzunalioglu, Critical national infrastructure reliability modeling and analysis, Bell Labs Technical Journal 11 (3) (2006) 57–71.

[55] S. Walsh, S. Cherry, L. Roybal, Critical infrastructure modeling: An approach to characterizing interdependencies of complex networks & control systems, in: Human System Interactions, 2009. HSI '09. 2nd Conference on, 2009, pp. 637–641. doi:10.1109/HSI.2009.5091052.