Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Ευστάθεια Συστήματος με Αναδιανομή Φορτίου από την Οπτική του Επιτιθέμενου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΥΑΓΓΕΛΟΣ Τ. ΧΑΤΖΗΑΦΡΑΤΗΣ

Επιβλέπων : Αριστείδης Τ. Παγουρτζής
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Αύγουστος 2015

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Ευστάθεια Συστήματος με Αναδιανομή Φορτίου από την Οπτική του Επιτιθέμενου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΥΑΓΓΕΛΟΣ Τ. ΧΑΤΖΗΑΦΡΑΤΗΣ

Επιβλέπων : Αριστείδης Τ. Παγουρτζής
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21η Αυγούστου 2015.

....	....	....
Αριστείδης Παγουρτζής Αν. Καθηγητής Ε.Μ.Π.	Δημήτριος Φωτάκης Επίκ. Καθηγητής Ε.Μ.Π.	Μιχάλης Λουλάκης Επίκ. Καθηγητής Ε.Μ.Π.

Αθήνα, Αύγουστος 2015

....

Ευάγγελος Τ. Χατζηαφράτης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

System Fragility under Load Redistribution from an Attacker's Perspective

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΥΑΓΓΕΛΟΣ Τ. ΧΑΤΖΗΑΦΡΑΤΗΣ

Επιβλέπων : Αριστείδης Τ. Παγουρτζής
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Αύγουστος 2015

Περίληψη

Η ευρωστία/ευστάθεια (robustness) συστημάτων μελετά το πόσο ανθεκτικά είναι τα συστήματα σε καταστάσεις όπου έχουμε δυσλειτουργίες σε ορισμένα τμήματά τους. Μελετά τη συμπεριφορά υπολογιστικών, ηλεκτρικών, ενεργειακών, μηχανικών συστημάτων, δικτύων κλπ. Είναι ιδιαίτερα σημαντικός κλάδος με πολλές εφαρμογές στο σύγχρονο κόσμο του Internet και των προσπαθειών που γίνονται για έξυπνα ενεργειακά συστήματα. Επίσης, έχει πολλές εφαρμογές στα λεγόμενα Διασυνδεδεμένα ή Αλληλεξαρτώμενα (Interdependent) Συστήματα.

Στην παρούσα εργασία μελετάμε το πρόβλημα της αντοχής δικτύων/συστημάτων σε επιθέσεις που καταστρέφουν κόμβους, η πτώση των οποίων μπορεί να συμβάλλει ή να προκαλέσει περαιτέρω αλυσιδωτές καταστροφές στους υπόλοιπους κόμβους του δικτύου. Ο κύριος λόγος της σύζευξης ανάμεσα σε διαφορετικούς κόμβους είναι κάποιου είδους εξάρτηση (πχ παροχή ρεύματος,) ή η αναδιανομή φορτίου που συμβαίνει όταν ένας κόμβος δέχεται επίθεση. Η αναδιανομή μπορεί να γίνει σε όλο το δίκτυο ή μόνο σε επιλεγμένους (γειτονικούς) κόμβους. Ορισμένα χαρακτηριστικά που αφορούν το παραπάνω πρόβλημα συνδέονται επίσης και με τη διάδοση ιών στο πλυθησμό, τα κοινωνικά δίκτυα και το γνωστό πρόβλημα της μεγιστοποίησης της επιρροής.

Αρχικά, παρουσιάζουμε μέρος από την υπάρχουσα έρευνα γύρω από την ευρωστία συστημάτων και τις διάφορες τεχνικές προσέγγισης του προβλήματος που έχουν αναπτυχθεί. Μελετάμε περιπτώσεις όπου οι αρχικές καταστροφές είναι τυχαίες (πχ μπορεί να οφείλονται σε μια φυσική καταστροφή), αλλά και περιπτώσεις όπου οι επιθέσεις είναι στοχευμένες (πχ τρομοκρατικές επιθέσεις). Στη συνέχεια, παρουσιάζουμε ένα νέο μοντέλο για την αντοχή του δικτύου και για τις αλυσιδωτές καταστροφές και αποδεικνύουμε την δυσκολία του να σχεδιάσει κανείς τη βέλτιστη επίθεση έχοντας περιορισμό στο πλήθος των επιθέσεων και στο μέγεθός τους. Παρουσιάζουμε ορισμένες άπληστες τεχνικές που είναι μεν διαισθητικές αλλά μπορεί στη χειρότερη περίπτωση να έχουν κακές επιδόσεις. Στη συνέχεια, περιοριζόμαστε σε περιπτώσεις συστημάτων που είναι πιο συνηθισμένες στη πράξη, εμφανίζουν λιγότερο παθολογικές συμπεριφορές και, συνεπώς, μπορεί κανείς να σχεδιάσει αποδοτικούς αλγορίθμους για τη βέλτιστη λύση. Στο τέλος, παρουσιάζουμε ορισμένα ενδιαφέροντα προβλήματα και ερωτήματα για μελλοντική έρευνα που προέκυψαν κατά τη σύνθεση της παρούσας εργασίας. s

Λέξεις κλειδιά

Αναδιανομή Φορτίου, Ευστάθεια, Ευρωστία Δικτύου, Κοινωνική Ευρωστία, Ασφάλεια, Αλληλεξαρτώμενα Συστήματα, Αλυσιδωτές Πτώσεις, Χωρητικότητα, Επιθέσεις Δικτύου, Απληστοι Αλγόριθμοι

Abstract

System Robustness and System Fragility are used when dealing with the problem of finding out how robust (resistant to failure) is a system in case some of its parts are not working as expected and are malfunctioning. System robustness is interested in the behaviour of computer systems, electrical and mechanical systems, energy systems, network security etc. It is an important area of research and it has many applications in today's extremely connected world of the Internet and in the efforts for constructing complex systems such as the smart energy grid etc. In addition, it is also important for the design and study of the so-called Interdependent Systems.

In this thesis, we study the problem of cascading failures: that is the situation in which we have an attack that destroys some part of the system and these failures may lead to further failures to other parts of the system. The main reason of this cascading effect and interconnection between different nodes are the possible node dependencies (e.g. power supply) or the load redistribution that is happening when an attacked node fails. The load redistribution may be equal among all remaining nodes or may be targeted to selected few of the remaining nodes (e.g. neighbouring nodes). Some of the aspects considered here for the load redistribution problem are also related to virus propagation, social networks and the well-known problem of influence maximization.

At the beginning of this thesis, we present recent research in the area of system robustness and the different techniques that have been developed and that are used to address the problem. We study cases where the initial failures are random (e.g. due to a natural disaster) or the failures are because of strategically planned attacks (e.g. terrorist attacks). We also present a new approach to system robustness based on loads and capacities with cascading failures and we give proofs for the hardness of finding the best strategy to attack the system, if we have constraints on the number of attacks and on the magnitude of these attacks. We present some greedy algorithms that may be intuitive, but are not guaranteed to give a good solution and may in fact have really bad behaviour. Afterwards, we continue by imposing extra constraints on the values of the system's loads and capacities in order to acquire less pathological cases that are most frequently seen in real situations and we give efficient algorithms that find the optimal solution from an attacker's perspective. Finally, we conclude this thesis by briefly describing the future work that needs to be done in order to address some interesting problems that we came across during the preparation of this work.

Key words

Load Redistribution, Network Robustness, Social Robustness, Interdependent Systems,
Cascading Failures, Security, Capacity, Network Attacks, Greedy Algorithms

Ευχαριστίες

Πέρασαν κιόλας πέντε χρόνια! Πέντε χρόνια στο Πολυτεχνείο όπου είχα την τύχη να συναντήσω καθηγητές και συμφοιτητές τους οποίους θαυμάζω και εκτιμώ για το μυαλό τους, το χαρακτήρα τους και τις ικανότητές τους. Μέσα από μαθήματα, συζητήσεις εντός και εκτός τάξης και, γενικότερα, συναναστροφή με αυτούς τους ανθρώπους, συνειδητοποίησα τι μου αρέσει, τι δε μου αρέσει και κατάλαβα πόσο ενδιαφέρον είναι να είσαι φοιτητής, να μαθαίνεις συνεχώς καινουρια πράματα, καινούριες απόψεις, καινουριες ιδέες.

Οι κόποι των πέντε αυτών ετών ολοκληρώνονται σήμερα με την παρουσίαση της διπλωματικής μου. Θα ήθελα λοιπόν να ευχαριστήσω θερμά τον επιβλέποντα της διπλωματικής μου κ. Παγουρτζή για την εμπιστοσύνη που μου έδειξε, το χρόνο που αφιέρωσε, την καθοδήγησή του καθώς και για τις γόνιμες συζητήσεις που είχαμε. Επίσης, ευχαριστώ πολύ τον κ. Ζάχο που είχε πάντα όρεξη να μας κάνει μάθημα ακόμα και σε καφενεια (Ya Cafe) σε καιρούς καταλήψεων, τον κύριο Φωτάκη για την αγάπη που μου καλλιέργησε για τους αλγορίθμους μέσα από την απίστευτη ατμόσφαιρα που δημιουργεί στο μάθημα και φυσικά, τον κύριο Λουλάκη για το ενδιαφέρον που μου έδειξε και τις ωραίες συζητήσεις που είχαμε. Τέλος, θα ήθελα να ευχαριστήσω θερμά τον κ. Γλύτση και τον κ. Ράπτη που με έκαναν να αγαπήσω πολύ τη Φυσική και μου εξηγούσαν με λεπτομέρεια ό,τι χρειάστηκα.

Φυσικά, το μεγαλύτερο ευχαριστώ θέλω να πω στην οικογένειά μου που με στηρίζουν σε ό,τι και αν κάνω. Στο μπαμπά που με φροντίζει και πάντα ό,τι και να γίνει θα του λέω ότι τρώω καλά και θα κοιμάται ήσυχος, στη μαμά που μας βάζει πάνω από όλα και μας βοηθά πάντα σε ό,τι χρειαστούμε και στον Αντρίκο για την βοήθειά του από όταν ήμουν μικρός, τα lolz του και τις ατέλειωτες ώρες ουσιαστικών/φιλοσοφικών συζητήσεων επί παντός επιστητού.

Τέλος, από τις πρώτες αυτές ευχαριστίες που γράφω δεν θα μπορούσαν να λείπουν οι φίλοι μου και οι στιγμές που έχουμε μοιραστεί. Φίλοι Καλοί (= ΦΚ) ευχαριστώ και συνεχίζουμε!

Ευάγγελος Τ. Χατζηαφράτης,

Αθήνα, 21η Αυγούστου 2015

Contents

Chapter 1

Introduction

1.1 Motivation

How is it that small initial shocks can cascade to affect or disrupt large systems that have proven stable with respect to similar disturbances in the past? Why do some books, movies, and albums emerge out of obscurity, and with small marketing budgets, become popular hits, when many a priori indistinguishable efforts fail to rise above the noise? Why does the stock market exhibit occasional large fluctuations that cannot be traced to the arrival of any correspondingly significant piece of information?

These phenomena are all examples of what economists call information cascades during which individuals in a population exhibit herd-like behaviour because they are making decisions based on the actions of other individuals rather than relying on their own information about the problem. Although they are generated by quite different mechanisms, cascades in social and economic systems are similar to cascading failures in physical infrastructure networks and complex organizations in that initial failures increase the likelihood of subsequent failures, leading to eventual outcomes that are extremely difficult to predict, even when the properties of the individual components are well understood. Not as newsworthy, but just as important as the cascades themselves, is that the very same systems routinely display great stability in the presence of continual small failures and shocks that are at least as large as the shocks that ultimately generate a cascade. Cascades can therefore be regarded as a specific manifestation of the robust yet fragile nature of many complex systems: a system may appear stable for long periods of time and withstand many external shocks (robust), then suddenly and apparently inexplicably exhibit a large cascade (fragile).

We are investigating the above questions since the quality of life in modern society strongly depends on the effective delivery of basic services such as water, electricity, and communications; the infrastructures providing these basic services are called critical infrastructures. Indeed, maintaining every critical infrastructure is a growing challenge for modern society. Of great interest is the interdependence of critical infrastructures and their robustness. One clear example of this interdependence is the binomial system in which electrical power networks depend on telecommunication networks and vice versa. Understanding cascading failure in interdependent networks is a problem currently receiving much attention. For example, failure propagation is a common phenomenon that can lead to such catastrophic effects as the remarkable September 2003 total blackout across Italy.

Cascading-like phenomena can be thought of as natural disasters as well. Sometimes a system fails because a part of it was put off due to a fire, a flood or other natural catastrophe. In this case, we might want to model natural disasters as random attacks to our network. In some other cases, like terrorists attacks there might be a strategy behind the attacks and so we might be interested as researchers in the optimisation version of robustness and cascading failures problems. By studying the problem of how systems evolve over time or can sustain external perturbations caused either by nature or an attacker is important for many reasons. First of all, we acquire knowledge that helps us understand *why* a system behaves like this.

For example, in the research paper of Marzieh Parandehgheibi and Eytan Modiano [17] they managed to explain using an appropriate system metric (Node-Minimum Total Failure Removals) why the 2003 blackout in Italy took place. They measured the Node-MTFR metric on the Italian network topology shown in [1]. For the power grid, they only considered the substations that are directly connected to the generators. These substations are the most critical nodes as if they fail, all dependent substations will fail. Similarly, for the CCN (Control and Communication Network), they only considered the routers that are directly connected to the control centers (they assumed that the control centers are located in the highly connected clusters of routers). Then, using the interdependency map in Italy they showed that for a total failure, one should hit particularly nodes in order to take out of all the cycles created with the interdependencies. The interdependency map revealed that the cycles were generally very short, and mostly isolated; therefore, one should remove a large number of nodes (13 nodes) simultaneously in order to cause the failure of the whole network and that was highly improbable. However, if one looked the northwest of Italy, he/she would see that one third of the substations are controlled by a few routers. In fact, removing only three routers will lead to blackout in one third of the power grid. This shows that although the network might be robust to a total failure, only a few node removals can cause a considerable partial blackout.

Another important aspect of studying robustness is that the designers of systems can have knowledge that will lead to better design decisions. What nodes should be important ? Where should the important nodes be placed? How should the remaining nodes be allocated ? Which part of the network is more vulnerable for failures ? What can we do to protect it? The answer to many of these questions lies in the research for system robustness.

An example is the research paper of Yagan, Qian, Zhang et al [36] in which they study a cyber-physical system consisting of two interacting networks, i.e., a cyber-network overlaying a physical-network. It is envisioned that these systems are more vulnerable to attacks since node failures in one network may result in (due to the interdependence) failures in the other network, causing a cascade of failures that would potentially lead to the collapse of the entire infrastructure. The robustness of interdependent systems against this sort of catastrophic failure hinges heavily on the allocation of the (interconnecting) links that connect nodes in one network to nodes in the other network. In this paper, they characterize the optimum inter-link allocation strategy against random attacks in the case where the topology of each individual network is unknown. In particular, they analyze the regular allocation strategy that allots exactly the same number of bi-directional inter-network links to all nodes in the system. With their research work designers of system can have a robust way of thinking about how to allocate the interconnections and the interdependencies, thus yielding to systems that can optimally or to some extent resist failures.

1.2 Preliminaries

In this section we give an overview of important definitions and results that will be useful in the next chapters.

1.2.1 Submodular & Supermodular Functions

The submodularity is useful in the context of social networks and robustness of networks. Sometimes we will show that the submodularity property holds and sometimes it does not. We will also present for completeness the supermodularity property.

Submodularity

Consider an arbitrary function $f(\cdot)$ that maps subsets of a finite ground set U to non-negative real numbers. We say that f is *submodular* if it satisfies a natural diminishing returns property: the marginal gain from adding an element to a set S is at least as high as the marginal gain from adding the same element to a superset of S. Formally, a submodular function satisfies $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$, for all elements v and all pairs of sets $S \subseteq T$. Submodular functions have a number of very nice tractability properties; the one that is relevant to us here is the following. Suppose we have a function f that is submodular, takes only nonnegative values, and is monotone in the sense that adding an element to a set cannot cause f to decrease: $f(S \cup \{v\}) - f(S) \geq 0$, for all elements v and sets S. We wish to find a k-element set S for which $f(S)$ is maximized. This is an NP-hard optimization problem (it can be shown to contain the Hitting Set problem as a simple special case), but a result of Nemhauser, Wolsey, and Fisher shows that the greedy hill-climbing algorithm that simply chooses the element that, for the moment, gives the maximum marginal gain approximates the optimum to within a factor of $(1 - 1/e)$ (where e is the base of the natural logarithm). The following algorithm: start with the empty set, and repeatedly add an element that gives the maximum marginal gain, is seen many times in the bibliography around sub modular functions and most recently (as of 2003) around the influence maximisation problem due to the seminal work of Kleinberg, Tardos et al. [2].

Supermodularity

We present the definition of supermodular functions. Consider an arbitrary function $f(\cdot)$ that maps subsets of a finite ground set U to non-negative real numbers. We say that f is *supermodular* if the marginal gain from adding an element to a set S is at most as high as the marginal gain from adding the same element to a superset of S. Formally, a supermodular function satisfies $f(S \cup \{v\}) - f(S) \leq f(T \cup \{v\}) - f(T)$, for all elements v and all pairs of sets $S \subseteq T$.

The above definitions can be extended for functions that take negative values as well, though in natural real life problems we are mostly interested in values that are non-negative.

1.2.2 Approximation Algorithms

We present an introduction to the main definitions used in the Approximation Algorithms bibliography [3], as it will prove useful later in chapter 5 in order to prove an inapproximability result. An approximation algorithm produces a feasible solution that is close to the optimal one, and is time efficient. The formal definition differs for minimization and maximization problems. Let Π be a minimisation (maximization) problem, and let δ be a function, from non-negative integers to non-negative rational numbers, with $\delta \geq 1 (\delta \leq 1)$. An algorithm A is said to be a factor δ approximation algorithm for Π if, on each instance I, A produces a feasible solution s for I such that $f_\Pi(I, s) \leq \delta(|I|) \cdot OPT(I)$ $(f_\Pi(I, s) \geq \delta(|I|) \cdot OPT(I))$ and the running time of A is bounded by a fixed polynomial in $|I|$. (f_Π denotes the objective function of the problem Π, OPT denotes the objective function value of an optimal solution to instance I). Clearly, the closer δ is to 1, the better is the approximation algorithm.

1.2.3 NP-complete Problems

Here we will focus on 2 different NP-Complete problems that are important for our analysis: Subset-Sum and Set Cover.

Subset Sum

In computer science, the Subset Sum problem is one of the important problems in complexity theory and cryptography. The problem is this: given a set (or multiset) of integers and a target sum S, is there a non-empty subset whose sum is S? For example, given the set $\{7, 3, 2, 5, 8\}, S = 0$, the answer is yes because the subset $\{3, 2, 5\}$ sums to zero. The problem is NP-complete. Here we present a famous variant of this problem which is still NP-complete. The original Subset Sum problems asks : "Given a set of integers and an target sum S, is there a non-empty subset whose sum is S?"; whereas the variant we are interested in asks : "Given a set of integers and a target sum S, is there a subset of **size k** whose sum is S?". We name this variant k-Subset Sum problem. For example, if $k = 1$, one can do a binary search to find the answer in $\mathcal{O}(\log n)$. If k = 2, then you can get the complexity down to $\mathcal{O}(n \log n)$ by fixing the first addendum of the sum and doing binary search for the remaining difference.

Proposition 1.2.1. *The k-Subset Sum problem is NP-complete.*

Proof. We will give a brief reduction from the original Subset Sum. First of all, the k-Subset Sum variant is obviously in NP since a "YES" certificate can be polynomially checked for correctness; that is, given the k numbers of the given set we can check in linear time that they add up to the target sum S. Secondly, imagine that there was an algorithm $F(\cdot)$ that has polynomial time complexity (in $n, \log S$) and could solve the decision problem of k-Subset Sum given as input the size k of the desired solution we are searching for. Then, the original decision problem of Subset-Sum would also be in P, as it could be represented as $F(1)|F(2)|...F(n)$ where F is our previous algorithm. This would have complexity of $\mathcal{O}(F(1) + F(2) + ... + F(n))$ which would still be polynomial; that means that we would have solved Subset-Sum in polynomial time, which is not true unless P=NP. $\square$

Set Cover

Another famous NP-complete problem is the Set Cover problem which is the following: Given a set of elements {1,2,...,m} (called the universe) and a set S of n sets whose union equals the universe, the set cover problem is to identify the smallest subset of S whose union equals the universe. For example, consider the universe $U = \{1, 2, 3, 4, 5\}$ and the set of sets $S = \{\{1, 2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}\}$. Clearly the union of S is U. However, we can cover all of the elements with the following, smaller number of sets: $\{\{1, 2, 3\}, \{4, 5\}\}$.

More formally, given a universe U and a family S of subsets of U, a cover is a subfamily $C \subseteq S$ of sets whose union is U. In the set covering decision problem, the input is a pair (U, S) and an integer k; the question is whether there is a set covering of size k or less. In the set covering optimization problem, the input is a pair (U, S), and the task is to find a set covering that uses the fewest sets. The decision version of set covering is NP-complete, and the optimization version of set cover is NP-hard.

1.2.4 Probability Distributions

In what follows, some basic knowledge of probability theory will be needed, especially for chapter 2, where we will discuss the robustness of a single system under random attacks. We will need 3 probability distributions: uniform, Pareto and Weibull. We briefly discuss these.

The probability density function of the continuous uniform distribution is:

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \le x \le b, \\ 0 & \text{for } x < a \text{ or } x > b \end{cases}$$

The values of f(x) at the two boundaries a and b are usually unimportant because they do not alter the values of the integrals of f(x) dx over any interval, nor of x f(x) dx or any higher moment. Sometimes they are chosen to be zero, and sometimes chosen to be $\dfrac{1}{b-a}$.

The Pareto distribution, named after the Italian civil engineer, economist, and sociologist Vilfredo Pareto, is a power law probability distribution that is used in description of social, scientific, geophysical, actuarial, and many other types of observable phenomena.

We denote by x_m the minimum value of the random variable X. With $b, x_m > 0$, we set

$$f(x) = x_m^b b x^{-b-1} \mathbf{1}[x \ge x_m] \tag{1.1}$$

To ensure that $E[X]$ is finite, we also enforce that $b > 1$; in that case we have $E[X] = bx_m/(b-1)$.

Finally, the Weibull Distribution has the form :

$$f(x) = \frac{k}{l}\Big(\frac{x - x_m}{l}\Big)^{k-1} e^{-(\frac{x-x_m}{l})^k} \mathbf{1}[x \ge x_m], \tag{1.2}$$

with $l, k > 0$. The case $k = 1$ corresponds to the exponential distribution, and $k = 2$ corresponds to Rayleigh distribution. The mean value is given by $E[X] = x_m + l\Gamma(1 + 1/k)$, where $\Gamma(\cdot)$ is the gamma-function.

1.3 Contribution of our Work

In chapter 5, we present the contribution of this thesis. We study the problem of cascading failures: that is the situation in which we have an attack that destroys some part of the system and these failures may lead to further failures to other parts of the system. The main reason of this cascading effect and interconnection between different nodes are the possible node dependencies (e.g. power supply) or the load redistribution that is happening when an attacked node fails. The load redistribution may be equal among all remaining nodes or may be targeted to selected few of the remaining nodes (e.g. neighbouring nodes). Some of the aspects considered here for the load redistribution problem are also related to virus propagation, social networks and the well-known problem of influence maximization.

At the beginning of this thesis, we present recent research in the area of system robustness and the different techniques that have been developed and that are used to address the problem. We study cases where the initial failures are random (e.g. due to a natural disaster) or the failures are because of strategically planned attacks (e.g. terrorist attacks). We also present a new approach to system robustness based on loads and capacities with cascading failures and we give proofs for the hardness of finding the best strategy to attack the system, if we have constraints on the number of attacks and on the magnitude of these attacks. We present some greedy algorithms that may be intuitive, but are not guaranteed to give a good solution and may in fact have really bad behaviour. Afterwards, we continue by imposing extra constraints on the values of the system's loads and capacities in order to acquire less pathological cases that are most frequently seen in real situations and we give efficient algorithms that find the optimal solution from an attacker's perspective. Finally, we conclude this thesis by briefly describing the future work that needs to be done in order to address some interesting problems that we came across during the preparation of this work.

1.3.1 Load Redistribution Problems

Our work is focused on the definition of two problems, namely $minCost$-ER-k and ERAN-k Problem (ER: Equal Redistribution, ERAN: Equal Redistribution Among Neighbors). The first is about finding a set of nodes, under a load-capacity model and under democratic load redistribution, that an attacker having limited budget might want to attack in order to destroy greater parts of the system. We prove that $minCost$-ER-k is NP-complete with a reduction from the k-Subset Sum variant presented in the Preliminaries above. The second is different in the sense that we have topology knowledge and the redistribution is happening only among neighbouring nodes and not democratically among all. Here we give a proof that it is NP-hard to approximate to any non-trivial approximation ratio with a reduction from the Set-Cover which was inspired by [2]. Then, we also proceed by giving variations of the equal-redistribution problems depending on the relations between the values of loads and the values of capacities. The model is powerful since we can model many realistic scenarios by giving loads and capacities the relations that are imposed by the real life scenario. For example, it might be the case that loads are the same but the capacities are arbitrary, or it might be the case that capacities are proportional to loads etc. In many of these variations we prove that a greedy approach can find the optimal solution.

Chapter 2

System Robustness under Probabilistic Failures and Load Redistribution

2.1 Introduction

In computer science, robustness is the ability of a computer system to cope with errors during execution. The meaning of robustness of a system, generally, reflects our perception of its reliability and its ability of tolerating perturbations that might affect its functional body. For example, one might think of the robustness of the Internet or the power grid and its importance in our lives which is highlighted in periods of system failures and large scale black-outs. A major concern that arises is how we can protect our system from these large scale and most often unexpected failures. Many of these failures are attributed to some small initial shock of the system getting escalated due to internal dependencies of the components of our system. This escalation may ultimately lead to the system's collapse. Therefore understanding the dynamics of failures can shed light to the successful development of infrastructures.

2.2 Brief Summary of Results

In this chapter, the basic model is motivated by the fact that some goods may need constant advertisement or may simply be in high demand constantly, e.g. Apple iPhone, Google Search Requests, Water etc. In this case, a company has N suppliers that provide the network with the digital/physical products. The contribution of each supplier to the total sales of the product is modelled by the random variables $L_1, L_2, ..., L_N$ that are independently drawn from an arbitrary distribution $F_L(x) = P[L \leq x]$. In other words, we can view the variables L_i as a measure of how much does the i^{th} supplier sells or how much load the supplier is handling when functioning normally. In order to model the ability of the suppliers to adopt to extra load caused by others suppliers' failures we introduce their capacity $C_i = (1 + \alpha)L_i$ with $\alpha > 0$ denoting the *tolerance* parameter. We can view the capacity C_i as the maximum load, e.g. google search requests, a supplier can handle without breaking down. Once the

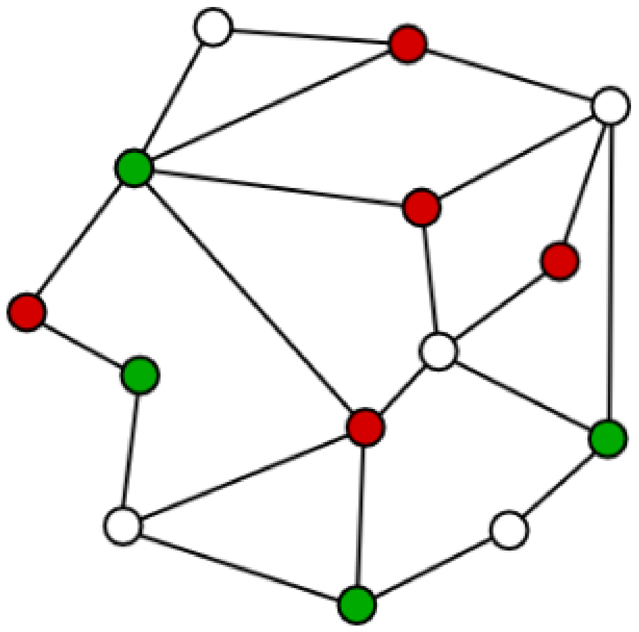


Figure 2.1: *For example this small graph could be part of a social market. The green nodes represent suppliers that are currently functioning. Red nodes are failed due to initial attacks or too much overload. We would be interested to see the propagation of failures in this network under load redistribution of the failed nodes.*

load of a supplier exceeds its capacity the supplier is under failure and cannot contribute at all to the company's sales.

When some of the N suppliers fail to deliver the good because of a malfunction, the company, that pursues the satisfaction of its customers via their unhindered service, will have to put pressure on the remaining well-functioning suppliers. We model this internal dependency by redistributing the load of work of these failed suppliers equally among the remaining live suppliers (democratic redistribution of work-load). This means that a supplier might need to work overtime; the load that has to deal with might increase over time due to failures of other suppliers.

In this chapter, the robustness of the system will be studied against random initial attacks (random failures); an initial $p - fraction$ of the suppliers will be attacked.The failure of the p-fraction of suppliers may cause further failures in the system due to the redistribution of their loads that may force some of the suppliers into exceeding their capacity. Subsequently, their load will be redistributed which in turn may cause further failures, and so on until the cascade of failures stops; note that this process is guaranteed to converge, at the very least when all product suppliers in the system fail.

An important result is that there is a critical threshold on the attack size p denoted p^*. Below p^* the suppliers that are functioning is a considerable fraction of N leading to complete fulfilment of customers' needs for the goods in the steady state, whereas above p^* the entire chain of suppliers collapses leading to zero sales in the steady state. In addition, the phase transition at p^* is always first-order; i.e., when we express the fraction of functional suppliers at the steady state as a function of the attack size p there is always a discontinuity of the first derivative. This means that small changes in p can lead to immense differences in our system's behaviour, rendering its robustness unpredictable. This phenomenon is well-known and is thought to be the origin of large but rare systems meltdowns seen in the real world, in a way explaining how small initial shocks can cascade to collapse large systems that have been proven stable with respect to similar disturbances in the past.

Another important result is that, depending on the distribution $F_L(x) = P[L \leq x]$ and the tolerance parameter α, our supply network can either collapse abruptly or can first have a preceding diverging rate of failures that serves as a warning message for the company to take action before the breakdown. In the former case, if $p < p^*$ the final fraction of alive suppliers will be given by $1 - p$ meaning that no single additional supplier fails other than those that are initially attacked, despite the fact that the whole supply system will suddenly collapse if the attack size exceeds p^*. In the latter case, we do not have an abrupt collapse without a warning: we have a preceding diverging rate of failures; that is, the final fraction of suppliers is less than $1 - p$ and this mere fact means that we are in a critical area where the supply system is about to collapse.

Finally, it is proven that the critical threshold p^* is maximized when the load distribution is a Dirac delta function centered at E[L], i.e., when all suppliers are equally working-selling. The optimal p^* is shown to be given by $\dfrac{\alpha}{\alpha + 1}$ regardless of the mean load E[L]. This finding is particularly surprising since heterogeneity is considered to be a useful tool when dealing with random failures (attacks). To this direction, it is proven that complex networks are known to be extremely robust against random failures when their degree distribution is broad [1]; e.g., when the number of links incident to a node in the network follows a power-law distribution.

2.3 Model Details

Our company has N suppliers providing our social market with goods. Each supplier has initial working/advertising load $L_1, L_2, ..., L_N$ that are independently drawn and identically distributed with distribution $F_L(x) = P[L \leq x]$. We assume that each supplier has positive workload which means that $L_{min} = sup\{x : P_L(x) = 0\} > 0$ since it would not make sense for an L_i to be negative. The probability density, given by $p_L(x) = \dfrac{d}{dx} P_L(x)$, is assumed to be continuous on its support. We also define the *capacity* of a supplier to be the maximum workload it can sustain. We assume that capacity $C_i = (1 + \alpha)L_i$ with $\alpha > 0$ denoting the *tolerance* parameter. Whenever the load that the supplier has to deal with exceeds his capacity, he fails and his load must be carried out equally by the remaining suppliers. The attack model is probabilistic meaning that there is an initial attack of volume p rendering useless (failed) a fraction p of the suppliers. In this setting, a cascade of failures is possible since each failure leads to extra burden for the remaining suppliers to deal with. For example, we can have N servers dealing with clients' search queries and as some of the servers are not functioning, the queries that were supposed to be answered by these failed servers must reroute to the well-functioning servers. If this extra burden is not leading to an excess in the capacity of a server then the system is stable and the company is fulfilling its clients' needs. However, if this extra overload leads to the breakdown of another server then we might have cascading failures of the servers.

2.4 Analytic Results

We will now describe the dynamics of the system in detail. We have the following mean-field analysis :

- Let f_t be the fraction of suppliers that are failed in time stage $t = 0, 1, ...$ Then the suppliers that are still functioning is obviously $N_t = N(1 - f_t), t = 0, 1, ...$
- Since the initial attack targets a p-fraction of the company's suppliers, we have $f_0 = p$. Their load of those $f_0 N$ suppliers will be redistributed among the $N(1 - f_0)$ live suppliers.
- Since the initial attack is random the extra load for each of the remaining suppliers Q_0 is given by
$$Q_0 = \frac{E[L]pN}{(1-p)N} = E[L]\frac{f_0}{1-f_0} \tag{2.1}$$
- A supplier i will not be able to address this extra overload and will fail if and only if:
$$C_i \leq L_i + Q_0 \Leftrightarrow (1+\alpha)L_i \leq L_i + Q_0 \Leftrightarrow L_i \leq Q_0/\alpha \tag{2.2}$$
- From the above inequality for L_i we expect an additional fraction of $P[L \leq Q_0/\alpha]$ suppliers to break down from overload. This means that at $t = 1$
$$f_1 = f_0 + (1-f_0)P[L \leq Q_0/\alpha] = f_0 + (1-f_0)(1 - P[L > Q_0/\alpha]) = 1 - (1-f_0)P[L > Q_0/\alpha]$$
Obviously, $f_1 \geq f_0$.
- In order to compute Q_1, i.e., the total load that will be redistributed per supplier at stage $t = 1$, we must sum the total load of all failed suppliers until now and divide with the current system size $1 - f_1$. This means that the new extra load per supplier Q_1 multiplied by the system size $(1 - f_1)$ makes for the total extra load the suppliers need to provide and this $Q_1(1 - f_1)$ is given by the sum of $Q_0(1 - f_0)$ and the total load of suppliers :
$$\begin{aligned} Q_1(1-f_1) &= Q_0(1-f_0) + E[\sum_{i \notin A_0 : L_i \leq Q_0/\alpha} L_i] \\ &= Q_0(1-f_0) + E[\sum_{i \notin A_0} L_i \mathbf{1}[L_i \leq Q_0/\alpha]] \\ &\overset{linearity\ of\ E}{=} Q_0(1-f_0) + \sum_{i \notin A_0} E[L_i \mathbf{1}[L_i \leq Q_0/\alpha]] \\ &\overset{E[x]:constant\ value}{=} Q_0(1-f_0) + (1-f_0)E[L\mathbf{1}[L \leq Q_0/\alpha]] \end{aligned}$$
Obviously, $Q_1 \geq Q_0$. We finally have :
$$Q_1 = \frac{pE[L] + (1-p)E[L\mathbf{1}[L \leq Q_0/\alpha]}{1-f_1} \tag{2.3}$$

- We note that $E[L\mathbf{1}[L \leq Q_0/\alpha]] = E[L|L \leq Q_0/\alpha]P[L \leq Q_0/\alpha]$

- To conclude the general form of f_{t+1}, Q_{t+1} we must observe that for a supplier to still stay alive at this stage, two conditions need to be satisfied: i) it should not have failed until this stage, which happens with probability $1 - f_t$ and ii) its load should satisfy $L > Q_t/\alpha$ so that its capacity is still larger than its current load. One additional note is that any supplier that satisfies condition (i) necessarily has a load $L > Q_{t-1}/\alpha$. Collecting, we obtain:

$$\begin{aligned} f_{t+1} &= 1 - (1 - f_t)P[L > Q_t/\alpha | L > Q_{t-1}/\alpha] \\ Q_{t+1} &= \frac{pE[L] + (1-p)E[L\mathbf{1}[L \leq Q_t/\alpha]}{1 - f_{t+1}} \\ N_{t+1} &= (1 - f_{t+1})N \end{aligned}$$

  We note that f_t, Q_t are monotone increasing as the time t goes by.

- For the cascade to stop we need $N_{t+1} = N_t$, or equivalently $f_{t+1} = f_t \Leftrightarrow$
  $\Leftrightarrow P[L > Q_t/\alpha | L > Q_{t-1}/\alpha] = 1$

2.4.1 Towards the Steady State

From the above equation for f_{t+1} we get :

$$\begin{aligned} 1 - f_{t+1} &= (1 - f_t)P[L > Q_t/\alpha | L > Q_{t-1}/\alpha] \\ 1 - f_t &= (1 - f_{t-1})P[L > Q_{t-1}/\alpha | L > Q_{t-2}/\alpha] \\ &. \\ &. \\ &. \\ 1 - f_1 &= (1 - f_0)P[L > Q_0/\alpha] \end{aligned}$$

Applying these recursively, we obtain :

$1 - f_{t+1} = (1 - f_0)\prod_{i=0}^{t} P[L > Q_i/\alpha | L > Q_{i-1}/\alpha]$,

where $Q_{-1} = 0$ for notation convenience . Since Q_t is monotone increasing in t, we can rewrite the above :

$$\begin{aligned} 1 - f_{t+1} &= (1 - f_0)\frac{P[L > Q_t/\alpha]}{P[L > Q_{t-1}/\alpha]}\frac{P[L > Q_{t-1}/\alpha]}{P[L > Q_{t-2}/\alpha]} \cdots \frac{P[L > Q_1/\alpha]}{P[L > Q_0/\alpha}P[L > Q_0/\alpha] = \\ &= (1 - f_0)P[L > Q_t/\alpha] \end{aligned}$$

Now, since $f_0 = p$, we can rewrite the expressions for Q_{t+1}, N_{t+1} a bit differently:

$$\begin{aligned} Q_{t+1} &= \frac{pE[L] + (1-p)E[L\mathbf{1}[L \leq Q_t/\alpha]}{(1-p)P[L > Q_t/\alpha]} \\ N_{t+1} &= (1-p)P[L > Q_t/\alpha]N \end{aligned}$$

To get to the steady state we must have $f_{t+2} = f_{t+1}$, which by substituting Q_{t+1} from above, is equivalent to :

$$P[L > Q_{t+1}/\alpha | L > Q_t/\alpha] = 1 \Leftrightarrow$$
$$\Leftrightarrow P\Big[L > \frac{pE[L] + (1-p)E[L\mathbf{1}[L \leq Q_t/\alpha]]}{\alpha(1-p)P[L > Q_t/\alpha]}\Big| L > Q_t/\alpha\Big] = 1$$

If we define $x := Q_t/\alpha$ we get the following more succinct condition for cascades of failure to stop in our market :

$$P[L > \frac{E[L] - (1-p)E[L\mathbf{1}[L > x]]}{\alpha(1-p)P[L > x]} | L > x] = 1 \tag{2.4}$$

It is now clear how to obtain the final fraction of functioning suppliers that are still alive at the end of the cascading failures: Since Q_t is increasing, as noted before, all we must do is find the smallest solution x^* of (4).This is why once our system reaches x^*, equation (4) will hold and, consequently, failures will stop. Then, the final fraction $n_\infty(p)$ of alive suppliers is given by:

$$n_\infty(p) = 1 - f_\infty = (1-p)P[L > x^*] \tag{2.5}$$

Let us investigate the equation (4). It holds in either one of the following two cases:

i) If $x \geq \dfrac{E[L] - (1-p)E[L\mathbf{1}[L > x]]}{\alpha(1-p)P[L > x]}$ or,

ii) If $x < \dfrac{E[L] - (1-p)E[L\mathbf{1}[L > x]]}{\alpha(1-p)P[L > x]}$ and $P\Big[L > \dfrac{E[L] - (1-p)E[L\mathbf{1}[L > x]}{\alpha(1-p)P[L > x]]}\Big] = 1$

In case, *ii)*, we observe that $P[L > x] = 1$ since:

$$x < \frac{E[L] - (1-p)E[L\mathbf{1}[L > x]]}{\alpha(1-p)P[L > x]} < L \tag{2.6}$$

This means that the final system size equals $1 - p$; no more suppliers other than the pN that were attacked initially will stop functioning. This is because of the equation (2) which in this case does not hold for any of the suppliers and so the additional fraction of suppliers that will fail because of the extra burden of the initially attacked suppliers is $P[L \leq Q_0/\alpha] = 0$. Substituting for Q_0 we get $P[L > \dfrac{pE[L]}{\alpha(1-p)}] = 1$, which can be regarded as the condition for no cascading of failures in the system. This condition can help in capacity provisioning by determining the factor α needed for robustness against $p-$size attacks. This condition can be rewritten :

$$L_{min} > \frac{pE[L]}{\alpha(1-p)} \tag{2.7}$$

Now let us investigate the first case *i)*.

$$x \geq \frac{E[L]-(1-p)E[L\mathbf{1}[L>x]]}{\alpha(1-p)P[L>x]} \Leftrightarrow P[L>x](\alpha x+E[L|L>x]) \geq \frac{E[L]}{1-p} \tag{2.8}$$

We can now see that the final system size $n_\infty(p)$ is always given by $(1-p)P[L > x^*]$ where x^* is the smallest solution of (8). This is clearly true for the case *i*) given above. To see why this approach also works for the case *ii*), observe that when (7) holds, (8) is satisfied for any x in $[\frac{pE[L]}{\alpha(1-p)}, L_{min}]$. Hence, the smallest solution x^* of (8) will always give $x^* \leq L_{min}$, leading to $(1-p)P[L > x^*] = 1-p$. As discussed before, no cascade takes place under (7) (i.e., in the case *ii*) above), so the final system size is indeed $1-p$.

For a graphical solution of $n_\infty(p)$, one shall plot $P[L > x](\alpha x + E[L|L > x])$ as a function of x (e.g., see Figure 1(a)), and draw a horizontal line at the height $\frac{E[L]}{(1-p)}$ on the same plot. The leftmost intersection of these two lines gives the operating point x^*, from which we can compute $n_\infty(p) = (1-p)P[L > x^*]$. When there is no intersection, we set $x^* = \infty$ and understand that $n_\infty(p) = 0$.

2.4.2 Total breakdown and early indicators

We calculated above the final system size $n_\infty(p)$ for a given attack size p. In many cases, we will be interested in the variation of $n_\infty(p)$ as a function of p since this will help us understand the response of the system to attacks of varying magnitude. Of particular interest will be to derive the critical attack size p^* such that for any attack with size $p > p^*$, the system undergoes a total breakdown leading to $n_\infty(p) = 0$

From (8) and the discussion that follows, we see that the maximum attack size p^* is related to the global maximum of the function $P[L > x](\alpha x + E[L|L > x])$. In fact, it is easy to see that

$$p^* = 1 - \frac{E[L]}{max_x P[L>x](\alpha x + E[L|L>x])} \tag{2.9}$$

The critical point x^* that maximizes the function $P[L > x](\alpha x + E[L|L > x])$ can shed light on the type of the transition that the system undergoes as the attack size increases. First of all, the system will always undergo a first-order (i.e., discontinuous) transition at the point p^*. This can be seen as follows: We have $n_\infty(p^*) = 0$by virtue of the fact that no x will satisfy (8), and cascading failures will continue until the whole system breaks down. On the other hand, $n_\infty(p^{*^-}) = (1-p)P[L > x^*] > 0$ where x^* is the point that maximizes $P[L > x](\alpha x + E[L|L > x])$. We can see why it must hold $P[L > x^*] > 0$ via contradiction: $P[L > x^*] = 0$ implies that the maximum value of $P[L > x](\alpha x + E[L|L > x])$ is zero, which clearly does not hold since at $x = 0$ this function equals E[L] ¿ 0 by non-negativity of L.

An interesting question is whether this first order rupture at the point p^* will have any early indicators at smaller attack sizes; e.g., a diverging failure rate leading to a non-linear decrease in $n_\infty(p)$. With $L_{min} > 0$, we know from (7) that for p sufficiently small, there will be no cascades and $n_\infty(p)$ will decrease linearly as $1-p$. This corresponds to the situations where (8) is satisfied at a point $x \leq L_{min}$, i.e., when $P[L > x](\alpha x + E[L|L > x])$ is linearly increasing with x. An abrupt first-order transition is said to take place if the linear decay of $n_\infty(p)$ is followed by a sudden discontinuous jump to zero at the point p^*. Those cases are

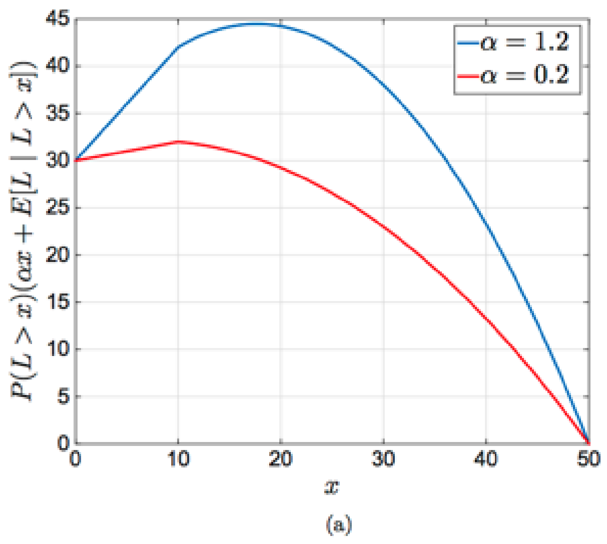
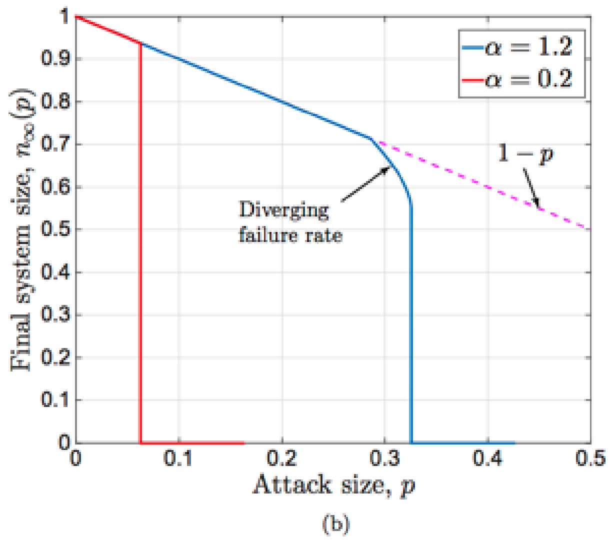


Figure 2.2: *We demonstrate the distinction between an abrupt first-order rupture, and a first-order rupture that is preceded by a diverging failure rate. In the first plot the straight lines are* $\alpha x + 30$ *right before the curved lines start. The market will always work at the left intersection point with the straight lines corresponding to constant numbers* $E[L]/(1-p)$*.* $p_L(x)$ *is assumed to be of uniform density over the range* $[L_{min}, L_{max}] = [10, 50]$*. In both plots, Red curves stand for the case where* $\alpha = 0.2$*, whereas Blue curves represent* $\alpha = 1.2$*. Figure 1(a) shows* $P[L > x](\alpha x + E[L|L > x])$*, whereas Figure 1(b) plots the corresponding variation of* $n_\infty(p)$ *with attack size* p*. We observe that for* $\alpha = 0.2$ *(Red),* $P[L > x](\alpha x + E[L|L > x])$ *takes its maximum at the point* $x = L_{min} = 10$*. As a result, we see an abrupt first-order transition of* $n_\infty(p)$ *as it suddenly drops to zero at the point* $p = p^* = 0.0625$*, while decaying linearly as* $1 - p$ *up until that point. The case where* $\alpha = 1.2$ *is clearly different as* $P[L > x](\alpha x + E[L|L > x])$ *is now maximized at* $x = 17.6 > L_{min}$*. As expected from our discussion, this ensures that the total failure of the system occurs after a diverging failure rate is observed. This divergence is clearly seen in Figure 1(b) where the dashed line corresponds to the 1 - p curve.*



reminiscent of the real-world phenomena of unexpected large-scale system collapses and can lead to our company's market demise; i.e., cases where seemingly identical attacks/failures leading to entirely different consequences. It is easy to see that an abrupt transition occurs if $P[L > x](\alpha x + E[L|L > x])$ takes its maximum at the point $x = L_{min}$; see Figure 1. In that case, (8) either has a solution at some xL_{min} so that $n_\infty(p) = 1 - p$, or has no solution leading to$n_\infty(p) = 0$. Under the assumptions enforced here, $P[L > x](\alpha x + E[L|L > x])$ is continuous at every $x \geq 0$, as a composition of continuous functions. Given that this function is linear increasing on the range $0 \leq x \leq L_{min}$, a maximum takes place at $x = L_{min}$ if at that point the derivate changes its sign. We have

$$\begin{aligned}
&\frac{d}{dx}(P[L > x](\alpha x + E[L|L > x])) \\
&= \frac{d}{dx}(\alpha x P[L > x] + E[L\mathbf{1}[L > x]]) \\
&= \alpha P[L > x] + \alpha x(-p_L(x)) + \frac{d}{dx}(\int_x^\infty t p_L(t)dt) \\
&= \alpha P[L > x] + \alpha x(-p_L(x)) - x p_L(x) \\
&= \alpha P[L > x] - x p_L(x)(\alpha + 1) \qquad (2.10)
\end{aligned}$$

where in the second to last step we used the Leibniz integral rule. As expected, for $x < L_{min}$, we have $P[L > x] = 1$and $pL(x) = 0$, so that the derivative is constant at α. For an abrupt rupture to take place, the derivative should be negative at the point $x = L_{min}$; i.e., we need

$$\alpha - L_{min} \cdot p_L(L_{min})(\alpha + 1) < 0$$

or, equivalently

$$\frac{\alpha}{(\alpha + 1)L_{min}} < p_L(L_{min}) \qquad (2.11)$$

It is important to note that (11) ensures only the existence of a local maximum of the function $P[L > x](\alpha x + E[L|L > x])$at the point $x = L_{min}$. This in turn implies that there will be a first order jump in $n_\infty(p)$ at the point where $E[L]/(1 - p) = \alpha L_{min} + E[L]$; i.e., at the point p that satisfies (7) with equality. However, for this condition to lead to an abrupt first-order breakdown, we need $x = L_{min}$ to be the global maximum. This can be checked by finding all x that make the derivate at (10) zero, and then comparing the corresponding maximum points. If $x = L_{min}$ is only a local maximum, then the system will have a sudden drop in size at the corresponding attack size, but will not undergo a complete failure; the complete failure and the drop of $n_\infty(p)$ to zero will take place at a larger attack size where, again there will be a first-order transition; e.g., see Figure 2.

We close by giving the general condition for first-order jumps to take place. We need a change of sign of the derivative at (10), leading to :

$$\pm\alpha P[L > x] - x p_L(x)(\alpha + 1)\Big|_{x = x^{*\pm}} < 0 \Leftrightarrow p_L(x^{*^-}) < \frac{\alpha P[L > x^*]}{(\alpha + 1)x^*} < p_L(x^{*^+}). \qquad (2.12)$$

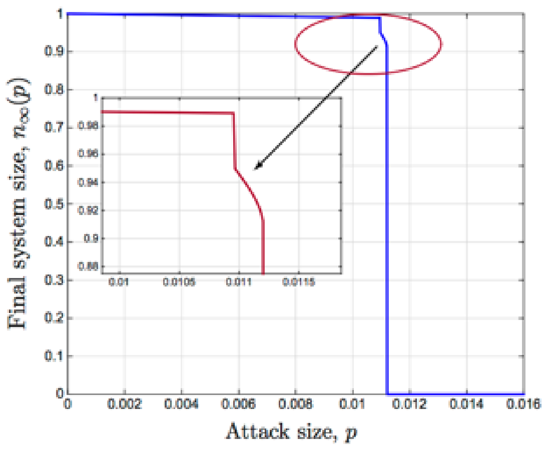


Figure 2.3: *The two stage breakdown of the system is demonstrated, where $L_1, ..., L_N$ are drawn from Weibull distribution with $k = 0.8, l = 150, L_{min} = 10, \alpha = 0.2$. We plot the relative final size $n_\infty(p)$ as a function of the attack size p. The Inset zooms in to the region where the system goes through a series of first-order, second-order, and then again a first-order transition.*

2.5 Suppliers with Specific Distributions

Now we will analyse 3 cases for the workload of the suppliers. In the first case we will let the L_i of the suppliers follow a Uniform distribution, in the second case follow a Pareto distribution and in the third case a Weibull distribution. We will compare them as far as the robustness of the social market they create and how stable or not it is under the probabilistic failures and the load redistribution model we saw earlier. (see FIG. 3)

2.5.1 Uniform Distribution

Assume that loads $L_1, ..., L_N$ are uniformly distributed over $[L_{min}, L_{max}]$. In other words, we have

$$p_L(x) = \frac{1}{L_{max} - L_{min}} \mathbf{1}[L_{min} \leq x \leq L_{max}], \tag{2.13}$$

so that

$$P[L > x] = \frac{L_{max} - x}{L_{max} - L_{min}} \mathbf{1}[L_{min} \leq x \leq L_{max}] + \mathbf{1}[x < L_{min}] \tag{2.14}$$

We see that over the range x in $[0, L_{max})$, the derivative of $P[L > x](\alpha x + E[L|L > x])$ (see (10)) is either never zero or becomes zero only once at

$$x^* = \frac{\alpha}{2\alpha + 1} L_{max}, \tag{2.15}$$

For the latter to be possible, we need $\frac{\alpha}{2\alpha+1} L_{max} \geq L_{min}$. If the opposite condition holds, i.e., if $\frac{\alpha}{2\alpha+1} L_{max} < L_{min}$, then $P[L > x](\alpha x + E[L|L > x])$ is maximized at $x = L_{min}$, and an **abrupt** first order break down will occur (as p increases) without any preceding

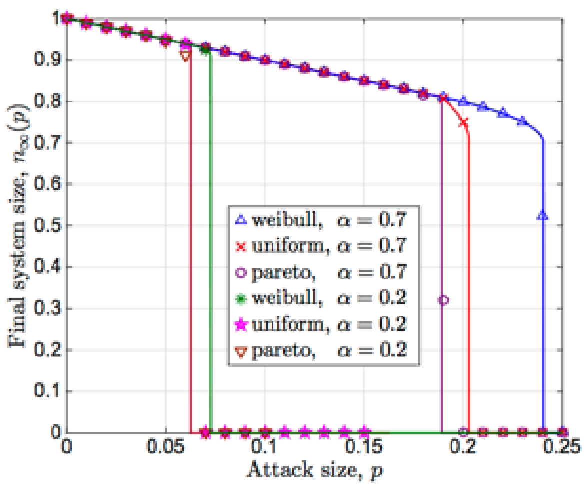


Figure 2.4: *We plot $n_\infty(p)$ vs. p under six different cases. Analytical results are represented by lines, whereas empirical results are represented by symbols. We set $N = 100,000, L_{min} = 10$, and $E[L] = 30$. For the case when $L_1, ..., L_N$ follow a Weibull distribution, we take the shape parameter to be $k = 2$, leading to a scale parameter $l = 22.5676$. We see that numerical results match the analytical results very well and so our mean field analysis is totally justified.*

diverging failure rate. As expected, the condition $\frac{\alpha}{2\alpha+1}L_{max} < L_{min}$ equivalent to the general rupture condition (11) and can be written most compactly as

$$\alpha < \frac{L_{min}}{max\{L_{max} - 2L_{min}, 0\}} \tag{2.16}$$

It follows that if $L_{max} \leq 2L_{min}$, then an abrupt rupture takes place irrespective of the tolerance factor α .

2.5.2 Pareto Distribution

Distribution of many real world variables are shown to exhibit a power-law behaviour, with very large variability (degree distributions in social networks, acquaintances in real-world etc). It would, thus , be of interest to consider social markets where the initial loads of the suppliers can exhibit high variance, we consider the case where $L_1, ..., L_N$ are drawn from a Pareto distribution: Namely, with $b, L_{min} > 0$, we set

$$p_L(x) = L_{min}^b b x^{-b-1} \mathbf{1}[x \geq L_{min}] \tag{2.17}$$

To ensure that $E[L]$ is finite, we also enforce that $b > 1$; in that case we have $E[L] = bL_{min}/(b-1)$. Then, the condition for an abrupt first order rupture (11) gives

$$\frac{\alpha}{(\alpha+1)L_{min}} < L_{min}^b b L_{min}^{-b-1} \tag{2.18}$$

or, equivalently $\alpha/(\alpha+1) < b$. With $b > 1$, this always holds meaning that when the suppliers' loads are Pareto distributed, there will always be an abrupt first order rupture at the attack size

$$p^* = 1 - \frac{E[L]}{E[L] + \alpha L_{min}} \tag{2.19}$$

In fact, we can see that this attack will lead to a complete breakdown of the system since for $x \geq L_{min}$, we have (see eq.(10))

$$\begin{aligned} &\frac{d}{dx}(P[L > x](\alpha x + E[L|L > x])) \\ &\quad = \alpha P[L > x] - x p_L(x)(\alpha + 1) \\ &\quad = \alpha \frac{L_{min}^b}{x^b} - (\alpha+1) x L_{min}^b b x^{-b-1} \\ &\quad = \frac{L_{min}^b}{x^b}(\alpha - b(\alpha+1)) < 0, \end{aligned}$$

for any $\alpha > 0$ and $b > 1$. Therefore, it is always the case that $P[L > x](\alpha x + E[L|L > x])$ has a unique maximum at $x = L_{min}$, and the abrupt first order rupture completely breaks down the system. These results show that for a given L_{min} and $E[L]$ with $E[L] > L_{min}$, Pareto distribution is the worst possible scenario in terms of the overall robustness of the market system. Put differently, with L_{min} and $E[L]$ fixed, the robustness curve $n_\infty(p)$ for the Pareto distribution constitutes a lower bound for that of any other distribution. From a design perspective, we see that changing the tolerance parameter α will not help in mitigating the abruptness of the breakdown of the system in the case of Pareto distributed loads. On the other hand, the point at which the abrupt failure takes place, i.e., the critical attack size p^* can be increased by increasing α.

2.5.3 Weibull Distribution

The Weibull Distribution has the form :

$$p_L(x) = \frac{k}{l}\Big(\frac{x - L_{min}}{l}\Big)^{k-1} e^{-(\frac{x-L_{min}}{l})^k} \mathbf{1}[x \geq L_{min}], \tag{2.20}$$

with $l, k > 0$. The case $k = 1$ corresponds to the exponential distribution, and $k = 2$ corresponds to Rayleigh distribution. The mean load is given by $E[L] = L_{min} + l\Gamma(1 + 1/k)$, where $\Gamma(\cdot)$ is the gamma-function. As usual, we check the derivative of $(P[L > x](\alpha x + E[L|L > x]))$ for $x \geq L_{min}$. On that range, we have $P[L > x] = e^{-(\frac{x-L_{min}}{l})^k}$ so that :

$$\frac{d}{dx}(P[L > x](\alpha x + E[L|L > x])) = e^{-(\frac{x-L_{min}}{l})^k}\Big(\alpha - (\alpha+1)x\frac{k}{l}(\frac{x - L_{min}}{l})^{k-1}\Big), \tag{2.21}$$

which becomes zero if

$$x(x - L_{min})^{k-1} = \frac{\alpha l^k}{(\alpha+1)k} \tag{2.22}$$

This already prompts us to consider the cases $k < 1$ and $k > 1$ separately. In fact, with $k > 1$, we see that $p_L(L_{min}) = 0$ and (11) does not hold regardless of α. In addition, there is one and only one $x > L_{min}$ that can satisfy (22). Consequently, for $k \geq 1$ the system will always undergo a second-order transition with a diverging rate of failure before breaking down completely through a first-order transition. The case $k < 1$ gives an entirely different picture since $p_L(L_{min}) = \infty$ and (11) always holds regardless of α. So, the system will always go through an abrupt first order transition at the attack size $p^* = 1 - \frac{E[L]}{E[L]+\alpha L_{min}}$. Whether this rupture will entirely breakdown the system depends on the existence of the solutions of (22). It is easy to see that $x(x - L_{min})^{k-1}$ takes its minimum value at $x = L_{min}/k$ and equals to $\frac{L^k_{min}}{k}(\frac{1-k}{k})^{k-1}$. Thus, if it holds that

$$L^k_{min}(\frac{1-k}{k})^{k-1} > \frac{\alpha l^k}{\alpha + 1}, \tag{2.23}$$

then (22) has no solution and the derivative given at (21) is negative for all $x \geq L_{min}$, meaning that $P[L > x](\alpha x + E[L|L > x])$ is maximized at $x = L_{min}$. Then, the abrupt first order rupture at $p^* = 1 - \frac{E[L]}{E[L]+\alpha L_{min}}$ will indeed breakdown the system completely. The same conclusion follows if (23) holds with equality by virtue of the fact that (21) is again non-positive for all $x \geq L_{min}$. On the other hand, if :

$$L^k_{min}(\frac{1-k}{k})^{k-1} < \frac{\alpha l^k}{\alpha + 1}, \tag{2.24}$$

then (22) will have two solutions both with $x > L_{min}$. This implies that $P[L > x](\alpha x + E[L|L > x])$ has another maximum at a point $x > L_{min}$. If this maximum is indeed the global maximum (i.e., it is larger than the maximum attained at $x = L_{min}$), then the system will go under two first-order phase transitions before breaking down. First, an abrupt rupture will take place at $p^* = 1 - \frac{E[L]}{E[L]+\alpha L_{min}}$. But, this wont break down the system completely and $n_\infty(p^{*^+})$ will be positive. As p increases further, we will observe a second-order transition with a diverging rate of failure until another first order rupture breaks down the system completely. We demonstrate this phenomenon in Figure 2, where we set $k = 0.8, l = 150, L_{min} = 10, \alpha = 0.2$. We emphasize that this behaviour (i.e., occurrence of two first-order transitions) is not immediately warranted under (24). It is also needed that $P[L > x](\alpha x + E[L|L > x])$ has a global maximum at a point $x > L_{min}$.

2.6 Optimal Load Distribution

As we observed by the analytic results for the above distributions, Pareto leads to the worst robustness, whereas Weibull distribution can lead to a significantly better robustness than Pareto and Uniform distribution, under the same mean and minimum load. The last observation worths investigating further. In particular, even when L_{min} and $E[L]$ are fixed, the Weibull distribution has another degree of freedom; i.e., parameters k and l are arbitrary subject to the condition that $l\Gamma(1 + 1/k) = E[L]$. In Figure 4 we plot our analytic results for the remaining nodes that are alive for different values of the parameter k. Specifically,

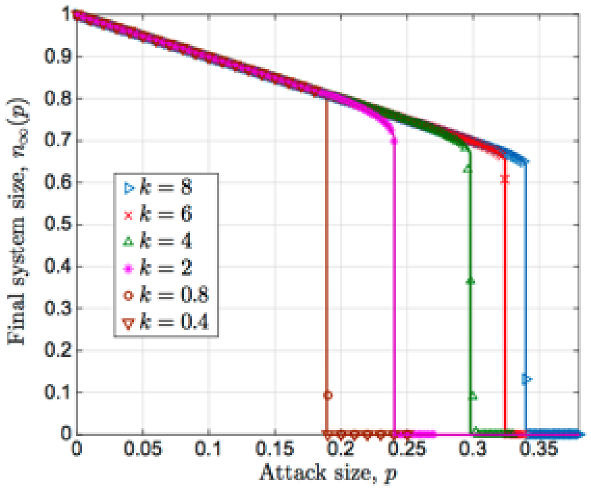


Figure 2.5: *We plot $n_\infty(p)$ vs. p when $L_1, ..., L_N$ follow a Weibull distribution with $L_{min} = 10, E[L] = 30$. We set $N = 100, 000$ and $\alpha = 0.7$. Analytical results are represented by lines, whereas empirical results are represented by symbols. Again, we see that numerical results match the analytical results pretty well. As k grows we observe improvements in the robustness of the market as the threshold of p^* moves to the right*

we let $\alpha = 0.7, L_{min} = 10, E[L] = 30$, and $l = 20/\Gamma(1 + 1/k)$ and we let k take the values written on the plot.

We see that the robustness of the system improves as the parameter k increases. It is known that as k gets larger the Weibull distribution gets closer and closer to a Dirac delta distribution centered at its mean. In other words, as k goes to infinity the Weibull distribution converges to a degenerate distribution and loads $L_1, ..., L_N$ will all be equal to the mean $E[L]$. This naturally prompts us to ask whether a degenerate distribution of loads is the universally optimum strategy among all possible distributions with the same mean $E[L]$, with optimality criterion being the maximization of robustness against random attacks or failures. Here, a natural condition for maximization of robustness would be to maximize the critical attack size p^*. We answer this question, in the affirmative. To drive the above point further and to better understand the impact of the shape parameter k on the system robustness, we now plot the maximum attack size p^* as a function of k under the same setting; see Figure 5. Namely, we let $L_1, ..., L_N$ follow a Weibull distribution with $L_{min} = 10$, and $l = 20/(1+1/k)$ so that $E[L] = 30$. We see from Figure 5 that, in all choices of α considered here, the maximum attack size p^* is monotone increasing with k; note that p^* is seen to be constant over the range $0 < k \leq 1$. It is also evident from Figure 5 that p^* tends to converge to a fixed value as k goes to ∞.

On the other hand, as k goes to ∞, we know that Weibull distribution converges to a Dirac delta distribution centered at E[L]. It is therefore of interest to check whether p^* is always maximized by choosing all loads$L_1, ..., L_N$ equally, i.e., by choosing $p_L(x)$ to be a degenerate distribution with mean E[L] and zero variance. Let $p_L(x)$ be an arbitrary distribution with mean E[L], and assume that $p_L(x) = 0$ for $x \leq 0$; i.e., that L is nonnegative. Recall that

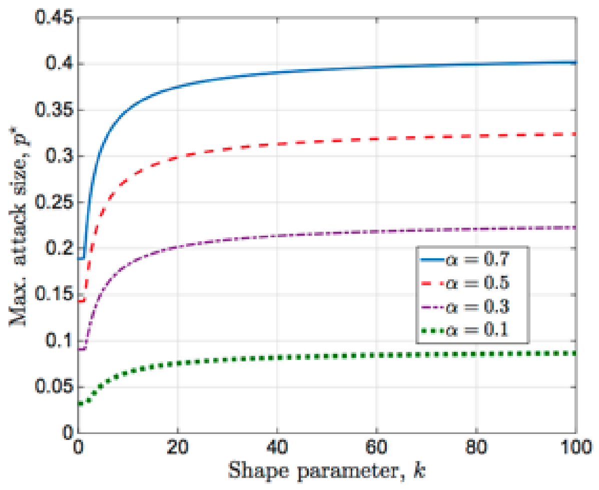


Figure 2.6: *We plot the maximum attack size p^*, when $L_1, ..., L_N$ follow a Weibull distribution with $L_{min} = 10, E[L] = 30$, as a function of the shape parameter k of the Weibull distribution. We set $N = 100, 000$ and consider four tolerance parameters $\alpha = 0.1, 0.3, 0.5, 0.7$. The curves correspond to analytical results computed directly from (9).*

maximum attack size p^* is given by (9) and observe that :

$$\begin{aligned}(P[L > x](\alpha x + E[L|L > x]))\\(\alpha x P[L > x] + E[L\mathbf{1}[L > x]])\\\leq \alpha E[L] + E[L\mathbf{1}[L > x]])\\\leq (\alpha + 1)E[L]\end{aligned}$$

for any $x \geq 0$. We note that we used the Markov Inequality i.e., the fact that $P[L > x] \leq E[L]/x$ for any non-negative random variable L and $x \geq 0$. Reporting the above inequality into (9), we get :

$$p^* \leq 1 - \frac{E[L]}{(\alpha + 1)E[L]} = \frac{\alpha}{\alpha + 1} \tag{2.25}$$

This shows that the maximum attack size can never exceed $\frac{\alpha}{\alpha+1}$ under any choice of load distribution. On the other hand, consider the case where $p_L(x) = \delta(E[L])$ with $\delta()$ denoting a Dirac delta function. This implies that $L_1 = \quad = L_N = E[L]$. Let p^*_{dirac} denote the corresponding maximum attack size. With $x = E[L]^-$, we have $P[L > x] = 1$ and $E[L \cdot \mathbf{1}[L > x]] = E[L]$. Thus,

$$\lim_{x \uparrow E[L]} (\alpha x P[L > x] + E[L\mathbf{1}[L > x]]) = (\alpha + 1)E[L] \tag{2.26}$$

so that,

$$max_x P[L > x](\alpha x + E[L|L > x]) \geq (\alpha + 1)E[L]. \tag{2.27}$$

If we report the above to (9) we get :

$$p^*_{dirac} \geq \frac{\alpha}{\alpha + 1}. \tag{2.28}$$

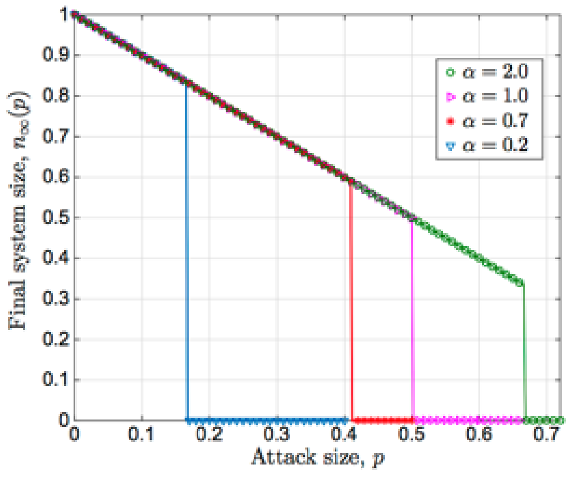


Figure 2.7: *We plot the final system size $n_\infty(p)$ as a function of the attack size p, when $L_1 = ... = L_N = E[L]$. For the numerical results, we take N = 100, 000, E [L] = 30, and consider four tolerance parameters $\alpha = 0.2, 0.7, 1.0, 2.0$. Numerical results are represented by symbols. The lines correspond to analytical results computed directly from (9). We see a perfect agreement between analysis and experiments. In all cases, the system breakdowns abruptly through a first order transition at $p^* = \alpha/(\alpha + 1)$.*

But, (25) holds for any distribution and hence is also valid for p^*_{dirac}.Combining these, we obtain that :

$$p^*_{dirac} = \frac{\alpha}{\alpha + 1} \tag{2.29}$$

This establishes that a degenerate distribution is indeed optimal for any given mean value of the load, and the achieved maximum attack size is given by $\frac{\alpha}{\alpha + 1}$. What is even more remarkable is that, this maximum attack size is independent of the mean load E[L]. It is now clear to what point the curves in Figure 5 tend to converge as $k \to \infty$; they can indeed be seen to get closer and closer to the corresponding value of $\frac{\alpha}{\alpha + 1}$. We close by demonstrating the variation of the final system size as a function of the attack size, in the case where loads follow a Dirac distribution. We easily see that $P[L > x](\alpha x + E[L|L > x])$ increases linearly for $x < E[L]$ and equals to zero for $x \geq E[L]$. Therefore, the breakdown of the system will always be through an abrupt first order rupture. This is demonstrated in Figure 6.

Comparing these plots with Figures 3 and 4, we see the dramatic impact that the load distribution has on the robustness of a market. For instance, with $\alpha = 0.2$ and mean load fixed at 30 we see that maximum attack size that the system can sustain is 6.3% for Pareto and Uniform distributions whereas it is 17% when all loads are equal. Similarly, with $\alpha = 0.7$ we see that maximum attack size is 18% for Pareto distribution and 19% for Uniform distribution, while for the Dirac delta distribution, it increases to 41%. These findings suggest that under the democratic fiber bundle-like model, where the load of the failed suppliers is equally distributed among the remaining alive suppliers, the social market with homogenous loads for the suppliers of the company is significantly more robust against random attacks and failures, as compared to other systems with heterogeneous load distribution.

2.7 Conclusion

We studied the robustness of a social market consisting of N suppliers under random attacks and redistribution of their load to the remaining suppliers. We show that the market goes under a total breakdown through a first-order transition as the attack size reaches a critical value. We derive the conditions under which the first-order rupture occurs abruptly without any preceding divergence of the failure rate; those situations correspond to cases where no cascade of failures occurs until a critical attack size. Last but not least, we prove that with mean load fixed, robustness of the market system is maximized when the variation among the suppliers' loads is minimized. In other words, a Dirac delta load distribution leads to the optimum robustness. Our results highlight how different parameters of the load distribution and the supplier capacity for extra work affect the robustness of the market against random failures and attacks. To that end, our results can help derive guidelines for the robust design of social markets. We believe that the results presented here give very interesting insights into the cascade processes in markets, although through a very simplified model of the society. The obtained results can be useful in other fields as well, where equal redistribution of flows is a reasonable assumption such as power systems or the Internet.

Chapter 3

Critical Node Failures in Interdependent Networks

3.1 Introduction

In this chapter, we present previous research in the area of the robustness of interdependent networks, in which the state of one network depends on the state of the other network and vice versa. The usual way of thinking the interdependent networks is having a communication network and the power grid; the communication network is dependent on the power grid, which in turn depends on the communication for control signals etc. However, we can view the two interdependent networks as being part of a wide social-advertising market. In this case, the one network might represent the advertisers that promote the products, whereas the second network is the companies that sell the products. The advertisers are dependent on the companies that pay them for their promotion and the companies are dependent on the advertisers in order to reach to the public and acquire clients. We want to investigate if a small failure in one of the networks can lead to major collapse of the system. We present the minimum number of node failures needed to cause total collapse (i.e., all nodes in both networks to fail). In the case of unidirectional interdependency between the networks we show that the problem is NP-hard, and develop heuristics to find a near-optimal solution. On the other hand, we show that in the case of bidirectional interdependency this problem can be solved in polynomial time. We believe that this new interdependency model gives rise to important, yet unexplored, robust network design problems for interdependent networked infrastructures.

3.2 Motivation for the problem

Nowadays infrastructure networks are interdependent in such a way that the failure of an element in one network may cascade to another network and cause the failure of dependent elements. This failure procedure can cascade multiple times between two or more interdependent networks and result in catastrophic widespread failures. A particular example is the interdependency between the power grid and communication networks; where the 2003 Italian blackout, affecting the lives of fifty-five million people, was the result of such interdependency. While reliability in networks has been studied extensively, the focus of most efforts has been on single networks in isolation. However, coupled networks exhibit very different behavior than isolated networks due to the cascading failure effects. In this paper, we develop a simple model for the interdependency between the networks, and analyze their robustness. There have been a few attempts at introducing and modeling interdependent infrastructure networks. Rinaldi et al. describe interdependencies among major infrastructure networks including the power grid and communication networks. In particular, they show that all infrastructure networks depend on information delivered by the communication network for monitoring and controlling their subsystems (i.e. SCADA). At the same time, they show that communication networks depend on the physical output of other networks; for example, power from the electric grid for switches or water from the water infrastructure for cooling systems. These interdependencies are referred to as cyber and physical, respectively. Later, Rosato et al. studied the Italian blackout of 2003 which affected the lives of fifty-five million people. They explain that this blackout was the result of a cascade of failures between the power grid and the communication network. In fact, due to failures in the power grid, some of the switches in the communication network lost their power and failed. Subsequently, due to failures in communication network, some of the substations in the power grid lost their control, and failed. They also demonstrate that failures inside the power grid lead to failures in the communication network using simulations on the real data of the Italian network. Although the papers by Rinaldi and Rosato explain the concept of interdependency between networks, neither presents a mathematical model for investigating the behavior of interdependent networks. Moreover, there has been some work on the interdependency within layered communication infrastructure, e.g. between fiber optical networks and IP networks, but, they did not consider interdependency between different infrastructures.

In 2009, Buldyrev et al. presented a model for analyzing the robustness of interdependent random networks. They generate two disjoint random graphs and define a one-to-one dependency between every pair of nodes in these graphs. Using techniques from percolation theory, they show that random failures cascade through the graphs, and investigate the existence of a giant component. Later, Parshani et al. showed that reducing the dependency between the networks makes them more robust to random failures.

However, real networks such as the power grid and the communication infrastructure, or the advertisers and selling stores networks are not random. These networks have known topologies; thus, it is essential to understand the impact of failure cascades in such networks in order to design robust interdependent infrastructures. In this chapter we will model **the interdependency between networks with known topologies**. The focus of this chapter is on the interdependency between power grid and communication networks; however, it can be extended to other networks with interdependency as well, such as the advertiser-selling networks discussed above.

We will now present the network model, define a new metric for assessing the robustness of networks and we will study the interdependency between networks with star topologies. Finally, we will evaluate the robustness of a real network and we will compare the results of different algorithms.

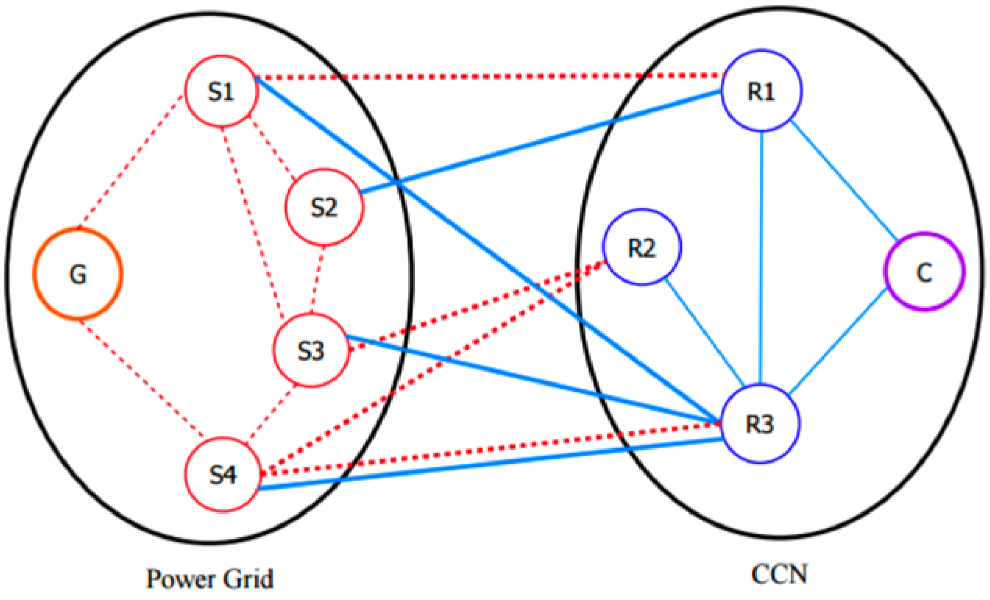


Figure 3.1: *For example, we have the power grid and the communication network with their dependencies. Dotted lines represent power lines and solid lines represent communication lines.*

3.3 The Interdependency Model

We start with a simple model for the power grid and the supporting control-communication network (CCN) . The power grid consists of generators G and substations S which are connected with power lines. Similarly, the CCN consists of control centers C and routers R which are connected with communication lines. Every router receives power from at least one substation and every substation sends data and receives control signals from at least one router. Figure 1 shows the interdependency model between the networks.

In this model, we say that a substation operates if it has a path to a generator, i.e. receives power, and it is also connected to a router, i.e. sends data and receives control signals. Similarly, we say that a router operates if it has a path to a control center, i.e. sends data and receives control signals and it is also connected to a substation, i.e. receives power. Thus, a failure in the power grid may cause a failure in the CCN and vice versa. We assume that the power generators have internal control, and the control centers have backup generators; thus, they are robust to failures. In addition, we also do not consider the amount of power supply or demand, and only focus on connectivity. Moreover, without loss of generality, we assume that there is only one generator and one control center; this is due to our models assumption that every substation can be connected to any generator, and every router can be connected to any control centers. Although this model is not fully realistic, it captures the essential properties of interdependent networks.

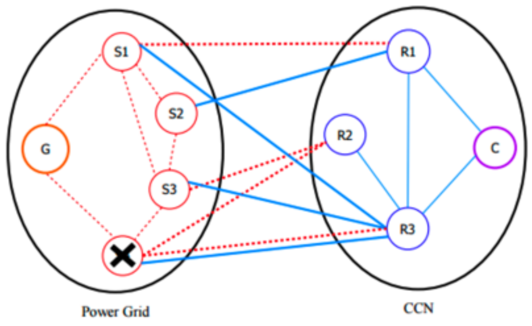
3.3.1 Effect of a Single Failure

We start with an example demonstrating that a single failure can cascade multiple times within and between the power grid and CCN (Figure 2). Suppose that initially substation S_4 fails (Step 1). As a result, all the edges attached to S_4 fail, and router R_3 loses its power and fails (Step 2); Consequently, substations S_1 and S_3 lose their control, and router R_2 loses its connection to the control center C, and all fail (Step 3).

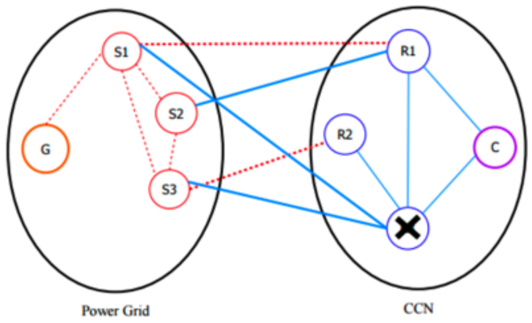
Finally router R_1 loses its power, and substation S_2 loses its connection to the generator G, and both fail (Step 4). In contrast, suppose that the power grid is not dependent on the CCN. In this case a substation fails, if and only if it is disconnected from the generator. For example, consider only the power grid from Figure 1 and, suppose that substation S_4 fails It can be seen that no other substation will be disconnected from the generator; thus, no further failure occurs. This example indicates that interdependency makes the networks more vulnerable, while it is essential to their operation. It was seen in the example of Figure 2 that a single failure can lead to the failure of all nodes in both networks. We define the failure of all nodes in both networks as **Total Failure**. This observation leads to an important question. What is the minimum number of nodes whose removals will lead to total failure? We define the **minimum total failure removals (MTFR)** as a metric that helps to measure the robustness of a network, i.e. the larger the MTFR, the more robust the network. In particular, we consider two types of Node and Edge removals which we refer to as Node-MTFR and Edge-MTFR metrics. In the case of a single network $G = (V, E)$, the smallest set of nodes that can cause a total failure, i.e. disconnect all the nodes in V from source S, is the set of nodes directly connected to S. Consequently, a star topology is the most robust network topology, since all of the nodes in a star are directly connected to the source.

3.4 Interdependency between Networks with Star Topologies

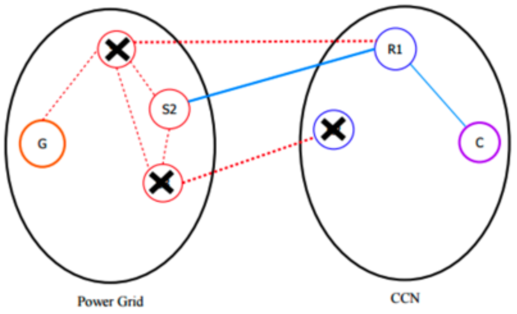
In order to analyze the interdependency between the power grid and the CCN, we assume that both networks have star topologies. Under this assumption, all of the substations in the power grid are directly connected to the generator; thus no substations failure can disconnect the other substations from the generator. Similarly, all of the routers in the CCN are directly connected to the control center, and no routers failure can disconnect the other routers from the control center. Therefore, any failure in the system would be only due to the interdependency between the networks, i.e. a substation fails if and only if it loses its connection to the CCN, and similarly a router fails if and only if it loses its connection to the power grid. This property gives us the opportunity to study the behavior of interdependent networks, and the effects of interdependency on the robustness of networks. Since both networks have star topologies, the interdependent network under study has a bipartite topology, i.e. no edge connects the nodes inside one network. In the following, we consider two distinct models of unidirectional and bidirectional interdependency. We analyze the networks under each model, and compare their complexity and robustness.



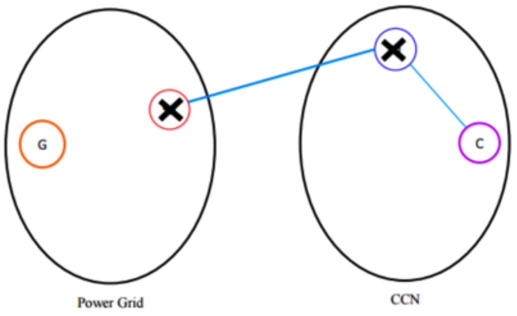
(a) Step1 - S_4 fails, initially



(b) Step2 - R_3 fails



(c) Step3 - S_1, S_3 and R_2 fail



(d) Step4 - R_1 and S_2 fail

Figure 3.2: *Cascade of a single failure in an interdependent model*

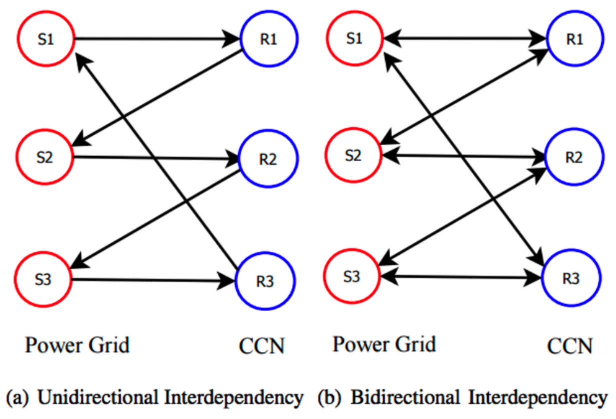

S1
S2
S3
R1
R2
R3
Power Grid
CCN
(a) Unidirectional Interdependency
S1
S2
S3
R1
R2
R3
Power Grid
CCN
(b) Bidirectional Interdependency

Figure 3.3: *Graph structure under different interdependency models*

3.4.1 *A.* Unidirectional Interdependency

Under the unidirectional interdependency model, the edges between the networks are directed, i.e. if a power node S_i provides power for a router R_j , the router R_j does not necessarily provide the control signal for the same power node S_i (Figure 3(a)). In the following we show that finding the Node-MTFR is in fact a hitting cycle problem, and prove that it is an NP-complete problem. In order to do so we start with the following lemmas:

Lemma 1: A network with one or more operating nodes has at least one cycle.

Proof. We prove by contradiction that no node in a network can operate if there is no cycle in the network. Suppose that there is no cycle, i.e. all nodes are connected through one or more paths. First, remove all of the nonoperating nodes; hence, the remaining nodes are operating, and the network is still acyclic. Now consider the starting node of one of the paths which is either a substation Si or a router Rj . This starting node does not have any incoming edges; therefore, substation Si (router Rj) does not receive any control (power) and cannot operate which is a contradiction with the assumption of nodes being operating. Now we show that existence of at least one cycle is sufficient to have an operating node. In a bipartite graph, every substation (router) in a cycle receives an incoming edge from a router (substation) in that cycle; thus, the nodes in that cycle can operate. If all the other nodes of the network receive an incoming edge directly or through a path starting from a node in that cycle, those nodes will be operating, too. □

Lemma 2: To stop the operation of any cycle, one of the nodes in the cycle should be removed (Cycles are Stable Components).

Proof. By definition, every substation (router) in the bipartite graph remains operating if it has an incoming edge from a router (substation). Every node in a cycle receives at least one incoming edge from the nodes inside that cycle; therefore, the removal of nodes outside the cycle will not affect the operation of the nodes inside that cycle. As a result, to stop the operation of a cycle, one of its nodes must be removed. □

Note that stopping the operation of a cycle is not equivalent to stopping the operation of all the nodes inside the cycle. If the nodes inside a cycle are isolated from the other nodes, removing exactly one node from the cycle will stop all of them from operating. However, if nodes inside a cycle receive incoming edges from other nodes outside of the cycle, more node removals are needed to cause the failure of all of the nodes in that cycle. ***Lemma 3:*** For total failure, at least one node from every cycle should be removed .

Proof. By contradiction - Suppose that there exist a cycle so that none of its nodes are removed. By lemma 2, all of the nodes inside that cycle remain operating, which contradicts the assumption of total failure. □

Theorem 1 The minimum number of nodes that hit all of the cycles in a bipartite graph is the optimal solution for the Node-MTFR problem.

Proof. Immediate from lemma 3. □

Corollary 1: Finding the Node-MTFR in star networks with unidirectional interdependency is NP-complete.

Proof. By Theorem 1, the Node-MTFR problem is a hitting cycle problem which is exactly equivalent to the wellknown problem of Feedback Vertex Set (FVS). By definition, FVS in a graph finds the smallest set of nodes so that their removals make the graph acyclic; and it is known to be NPcomplete for general graphs. Moreover, Cai et al. proved that FVS is NP-complete for a special class of bipartite graphs called bipartite tournament. Therefore, the Node-MTFR problem which is finding FVS in general bipartite graphs is also NP-complete. □

We now proceed with the problem formulation:

Problem Formulation: It was shown in lemma 3 that finding the Node-MTFR is a hitting cycle problem. Next, we present a cycle-based Integer Linear Programming (ILP) formulation for this problem, assuming that all of the cycles are given. Let N be a $n \times 1$ binary vector so that each component N_j takes values 1 if node j is removed and 0 otherwise. Let matrix $A \in R^{m \times m}$ be a mapping between the m cycles and n nodes, where $A_{ij} = 1$ if cycle i contains node j and $A_{ij} = 0$ otherwise. Let e be a $m \times 1$ vector of ones.
The Node-MTFR problem can be formulated as follows:

$$minimize \sum_{j=1}^{n} N_j \qquad (1)$$

$$subject\ to\ A \times N \geq e \qquad (2)$$

$$N_j \in \{0,1\}, j = 1, 2, .., n \qquad (3)$$

In this formulation, the objective is to minimize the number of node removals. Every row i of constraint (2) requires that at least one of the removed nodes should hit cycle i.
In the following, we develop heuristics to solve the problem.
Heuristics: Since computing the Node-MTFR is computationally difficult, we consider approximation algorithms that give a near-optimal set of node removals in polynomial time.

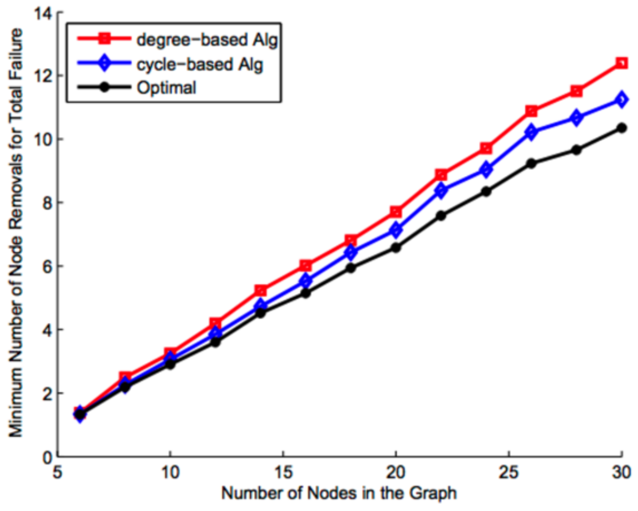


Figure 3.4: *Comparing different algorithms with optimal solution.*

As explained before, the node removals problem is equivalent to a hitting cycle problem. Thus, if we have the set of all cycles in the graph, we can apply a greedy algorithm devised for solving the hitting set problem. The input to the algorithm is the set of cycles (each cycle is defined as the set of nodes it contains), and the set of nodes in the graph. This cycle-based algorithm is an iterative algorithm that works as follows. In each iteration, it removes the node that is shared among maximum number of cycles, updates the set of cycles, and repeats until no cycle remains. This cycle-based algorithm needs the set of all cycles as input; however, in general, a graph may have an exponential number of cycles. To overcome this deficiency, we devise a new algorithm that relies on the degree of the nodes instead of the cycles. The input to the algorithm is the adjacency matrix of the graph. The algorithm is iterative: Each iteration starts with a pruning stage in which the algorithm removes all of the edges that do not belong to a cycle. In the next stage of the iteration, it removes the node that has the maximum outgoing degree. Next, the algorithm removes all nodes that fail as a result of the cascading effect of that removal. Finally, the algorithm updates the adjacency matrix of the graph and repeats the iteration until no node remains. In the following, we compare the performance of these algorithms with the optimal solution. We consider a random bipartite graph with N nodes on each side. Since enumerating all the cycles requires exponential time, we keep the size of N small, and limit the graph to have small cycles. To do that, instead of randomly generating edges, we randomly generate cycles of size 6 or smaller until all the nodes have at least one incoming edge. For each value of N, we generate 100 random graphs and then apply our algorithms to each graph in order to find the minimum node removals. Moreover, for the optimal solution, we solve the hitting cycle problem as given by (1)-(3) using the optimization software package CPLEX. As can be seen from Figure 4, on average, the degree-based algorithm gives a slightly larger number of nodes compared with the cycle-based algorithm and optimal solution; however, it is very fast as it does not need to enumerate all of the cycles.
A variation of the problem is described afterwards: *Minimum Edge Removals:* An alterna-

tive version of the problem is the minimum number of edges needed to be removed to cause a total failure. Similar to lemmas 2 and 3, to stop the operation of any cycle, one should remove one of its edges, and to have a total failure, at least one edge from every cycle should be removed. Consequently, we have the following results:

Theorem 2 The minimum number of edges that hit all of the cycles is the optimal solution for the Edge-MTFR problem.

Proof. Trivial. □

Corollary 2: Finding the minimum edge removals for total failure in the star networks with unidirectional interdependency is NP-complete.

Proof. By Theorem 2, minimum edge removals problem is the edge version of the hitting cycle problem which is exactly equivalent to the well-known problem of Feedback Edge Set (FES). Similar to FVS, FES finds the smallest set of edges whose removals make the graph acyclic, and it is known to be NP-complete for general graphs. Furthermore, Guo et al. proved that FES is NP-complete for bipartite tournaments. Since finding FES in bipartite tournaments is a special case of the Edge-MTFR problem, the Edge-MTFR problem is also NP-complete. □

3.4.2 *B.* Bidirectional Interdependency

Under the bidirectional interdependency model, the edges between the two interdependent networks are bidirectional, i.e. if a power node S_i provides power to router R_j , router R_j must provide control to power node S_i (Figure 3(b)).

Theorem 3 Finding the minimum node removals for total failure in star networks with bidirectional interdependency is solvable in polynomial time.

Proof. It is easy to see from Figure 3(b) that edges are cycles of length two, and hitting cycles of length two guarantees hitting cycles of larger size. On the other hand, hitting at least one node in every cycle of size two is equivalent to finding the minimum vertex cover in bipartite graphs. By Konigs Theorem, finding the minimum vertex cover in bipartite graph is equivalent to maximum matching which is polynomially solvable. Thus, finding the minimum node removals in star networks with bidirectional interdependency is polynomially solvable. □

For the Edge-MTFR, all of the edges must be removed. This is due to the fact that in the star networks with bidirectional interdependency every edge is a cycle, and cycles are robust components.

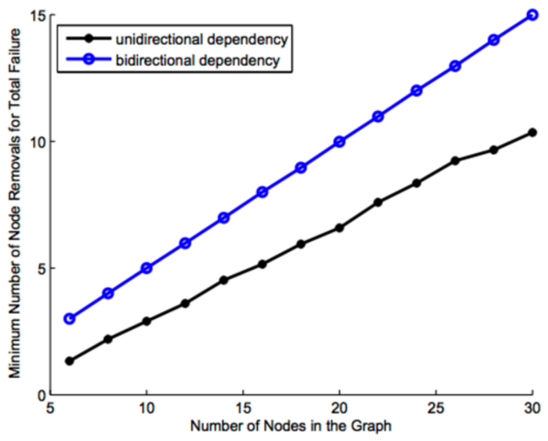


Figure 3.5: *Comparing the robustness of interdependency models.*

3.4.3 *C.* Comparing the Interdependency Models

We have seen that when networks have unidirectional interdependency, finding the optimal solution for the Node-MTFR problem is NP-complete; however, it can be solved in polynomial time when the networks have bidirectional interdependency. Here, we try to explain by way of an example why the analysis of unidirectional interdependency is more difficult than bidirectional interdependency. Figures 3(a) and 3(b) show two networks with the same topology under the different interdependency models. Suppose that in both networks, node S_1 is intentionally removed. It can be seen that the removal of S_1 in network 3(a) leads to the sequential failure of nodes R_1, S_2, R_2, S_3 and finally R_3. However, removal of S_1 in network 3(b) does not cause any failures, as all of the other nodes still have an incoming edge. Now, suppose that we want to cause the failure of node R_2 in network 3(b). In this case, all the nodes attached to R_2 (nodes S_1 and S_2) must be removed. It can be seen that failure of R_2 cannot cause the failure of other nodes as it is already disconnected from the rest of the network. These observations indicate that in the case of unidirectional interdependency, a failure can cascade multiple times between the networks. However, in the case of bidirectional interdependency, a failure cascades only in one stage: either from the power grid to the CCN or from the CCN to the power grid. This makes the analysis of bidirectional interdependent networks more tractable. Next we compare the robustness of the interdependency models. We use the random graphs generated before in order to compare our heuristics with the optimal solution and we generate a new set of graphs with the same topology but bidirectional dependency. To compare the robustness of the two models, we find the optimal solution in the unidirectional graphs by solving the hitting set problem using the software package CPLEX, and bidirectional graphs by solving the vertex cover problem. It can be seen from Figure 5 that for all values of N, networks with bidirectional dependency need more node removals; therefore, they are more robust to failures. This observation shows that the existence of more disjoint cycles and shorter cycles makes a network more robust.

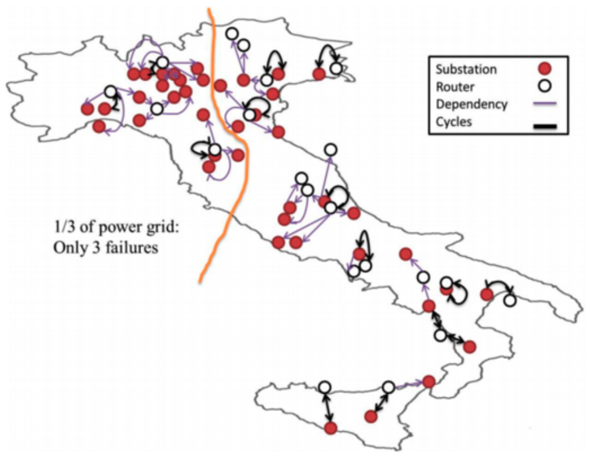


Figure 3.6: *The power network - communication network interdependence during the 2003 extensive black-out.*

3.5 Conclusion

We considered the problem of robustness in interdependent networks. Two networks A and B are said to be interdependent if the state of network A depends on the state of network B and vice versa. We considered a cyber-physical interdependency between the networks in the sense that the nodes in network A depend on information from network B, and the nodes in network B depend on the physical output of network A. The massive blackout in Italy in 2003 was the result of such interdependency between the power grid and the CCN where a small failure in the power grid cascaded between the two networks and led to extensive failures in both the power grid and the CCN.(see Figure 6)

We studied the minimum number of nodes that should be removed from both networks so that all of the nodes in the networks fail after the ensuing cascades. We formulated this problem, and proved it is equivalent to the hitting cycles problem which is NP-complete in the case of unidirectional dependencies. We also presented polynomial algorithms which give suboptimal solutions. Finally, for the case of bidirectional dependencies, we proved our problem is equivalent to the vertex cover problem in bipartite graphs which is solvable in polynomial time.

In the next chapter we will see a generalisation of the classical Vertex Cover Problem in interdependent networks for which we will also have the intra-topology; that is we will have knowledge of the edges connecting the nodes in each individual network.

Chapter 4

Vertex Cover and the Destruction of Interdependent Networks

4.1 Introduction

In the previous chapter, we presented previous research in the area of the robustness of interdependent networks but we did not have the topology knowledge of each network individually, only the dependencies were given. We now present a case in which the intra topologies are given.
In this chapter, we will define and analyze a generalization of the classical minimum vertex cover problem to the case of two-layer interdependent networks with cascading node failures that can be caused by two common types of interdependence. Previous studies on interdependent networks mainly addressed the issues of cascading failures from a numerical simulations perspective, whereas here we propose an exact optimization-based approach for identifying a minimum-cardinality set of nodes, whose deletion would effectively disable both network layers through cascading failure mechanisms. We analyze the computational complexity and linear 01 formulations of the defined problems, as well as prove an LP approximation ratio result that generalizes the well-known 2-approximation for the classical minimum vertex cover problem. In addition, we introduce the concept of a depth of cascade (i.e., the maximum possible length of a sequence of cascading failures for a given interdependent network) and show that for any problem instance this parameter can be explicitly derived via a polynomial-time procedure. Towards the end, we extend previous research by adding the load redistribution aspect to this problem and by proving hardness results.

First of all, an illustrative example of coupled interdependent networks is given in Fig. 1.

Interdependent links connect the nodes between the networks (only a few of such links are shown for illustrative purposes). Note that in general, multiple interdependent links may be outgoing from or incoming to a node, thus creating the possibility of one-to-many and many-to-one relationships between nodes in different networks, which generalizes the one-to-one relationships studied in previous models by Buldyrev et al. (2010).

Although quantitative analysis of network robustness can be addressed from a variety of perspectives, an intuitive and classical measure of robustness is the minimum number of

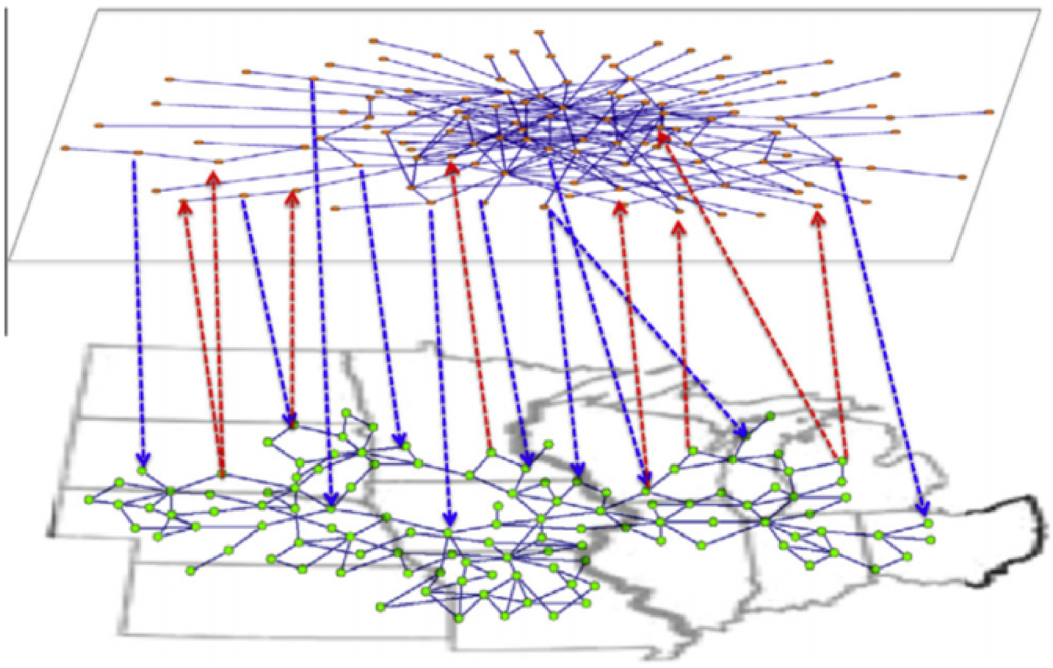


Figure 4.1: *Conceptual illustration of a two-layer (coupled) interdependent network.*

nodes (absolute, or relative to the size of the whole network) that need to be disabled in order to disrupt the functionality of the entire network. Obviously, if this number is small compared to the overall number of nodes in a network, then one can conclude that a network is vulnerable to attacks, whereas if this number is comparable with the size of the whole network, then such a network is more robust with respect to random or targeted disruptions. From an optimization perspective, if an adversary possesses complete information about the topology of a given network, he can solve an optimization problem of identifying a minimum-cardinality set of nodes that need to be disabled in order to limit the operational capabilities of the residual network. For instance, if the requirement is to limit the size of the remaining connected components by a pre-de- fined threshold L, such a problem is referred to as the cardinality-constrained critical node detection problem (Arulselvan, Commander, Shylo, & Pardalos, 2011; Veremyev, Boginski, & Pasiliao, 2013). In the special case of $L = 1$, this problem is reduced to the classical minimum vertex cover problem, which identifies a minimum-cardinality set of nodes that touches every edge in a network, thus resulting in a residual network with no edges after all nodes in the minimum vertex cover are disabled. Identifying the minimum-cardinality set of nodes whose deletion would effectively destroy the network is useful from both attack and defense perspectives. In the former setup, the attacker identifies an optimal set of nodes that need to be disabled, whereas in the latter setup, the defender has an opportunity to protect an identified set of nodes and potentially interfere with the propagation of cascading failures. Let us proceed with some basic definitions about the model presented here.

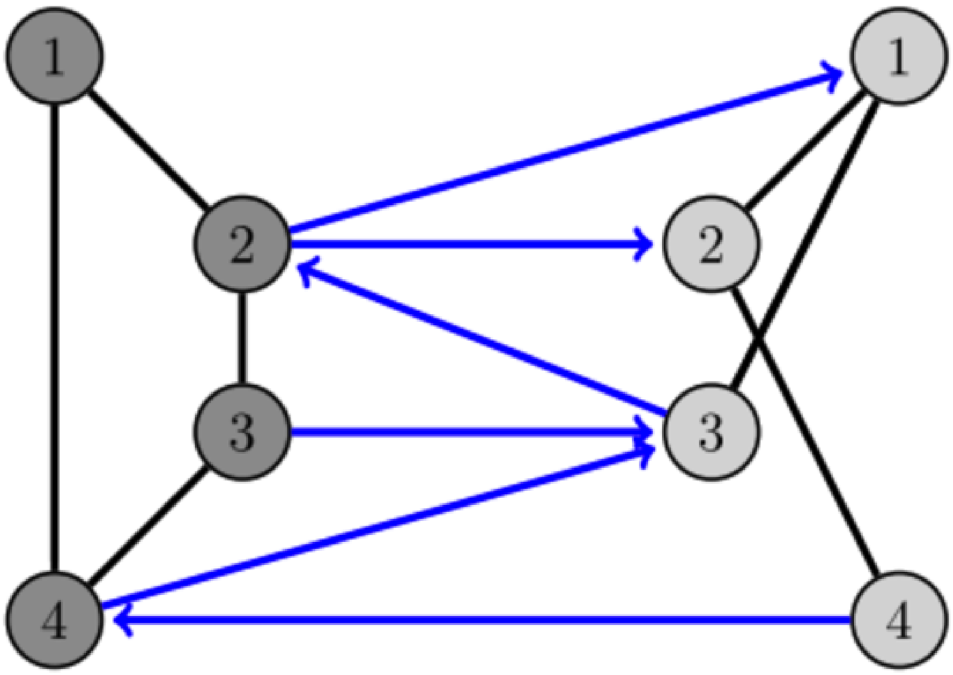


Figure 4.2: *An example of a 2-layer interdependent network.*

4.2 Cascading Failures: Basic Concepts

We model a coupled interconnected infrastructure as a two-layer interdependent network with graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ being its layers with sets of nodes $V_1 = \{1, ..., n_1\}, V_2 = \{1, ..., n_2\}$ $(n_1 + n_2 = n)$, and edges E_1, E_2. We assume that some nodes in one layer depend on nodes in the other layer, and denote by E_{12}, E_{21} the sets of interdependent links, meaning that if $(i, j) \in E_{12}$, then node $j \in V_2$ depends on node $i \in V_1$. For example, in the network depicted in Fig. 2: $E_{12} = \{(2, 1), (2, 2), (3, 3), (4, 3)\}$ and $E_{21} = \{(4, 4), (3, 2)\}$. For simplicity, we denote the entire coupled interdependent network as $G^{(2)} = ((V_1, E_1), (V_2, E_2), E_{12}, E_{21})$. We assume that the set of initially failed/attacked nodes causes subsequent failures of other nodes stage-by-stage due to interdependence (sets E_{12}, E_{21}), similar to the assumptions of Buldyrev et al. (2011). The motivation for this assumption is the fact that in many cases cascading failures do not spread instantaneously, but rather with some delay between cascade stages (Bienstock, 2011). We assume that the number of cascading stages is bounded by a fixed pre-specified parameter S. In this paper we consider two common types of interdependence: **Type 1 (one-to-many)** and **Type 2 (many-to-one)**.

Definition 1: An interdependent network has **Type 1** failure causality if the failure of **one** node causes the failure of **all** of its dependent nodes.

Definition 2: An interdependent network has **Type 2** failure causality if the failure of **one** node is caused only by the failures of **all** nodes it depends on.

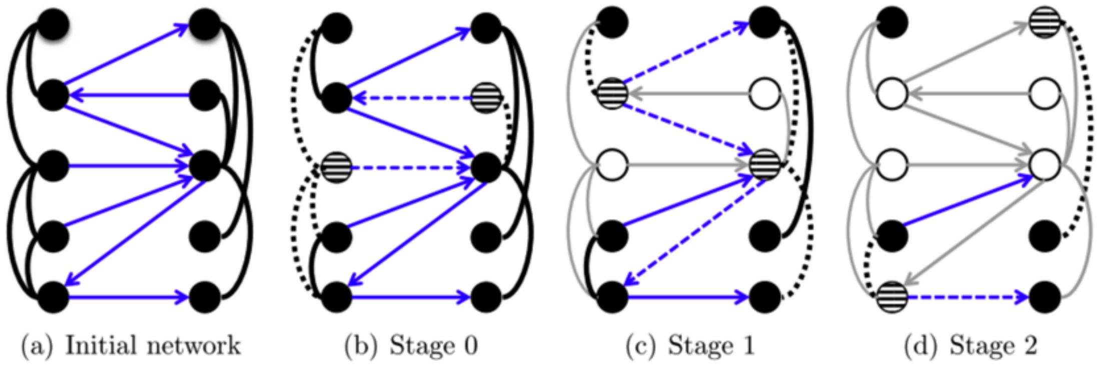


Figure 4.3: *Illustration of stage-by-stage node failures caused by Type 1 interdependence.*

For any given interdependent network $G^{(2)}$, type of interdependence and the number of cascading stages S, the stages of cascade and the vertex cover are defined as follows.

Definition 3: *(Cascade Stage:)* Let the initial set of failed/attacked nodes belong to *stage 0*. Then for any $s \in \{1, ..., S\}$, *stage s* contains nodes that either belong to *stage* $(s-1)$, or have been failed due to the dependence on nodes in *stage* $(s-1)$.

Definition 4: *(Vertex Cover:)*. A vertex cover of $G^{(2)}$ is a set of nodes in $V_1 \cup V_2$ whose failure causes the failure of vertex cover sets in graphs G_1 and G_2 at stage S through the corresponding cascading failure mechanisms.

For the given interdependent network $G^{(2)}$, type of interdependence and the number of cascading stages S the minimum vertex cover problem asks for the minimum cardinality vertex cover in $G^{(2)}$. Fig. 3 illustrates stage-by-stage node failures caused by disabling two nodes at stage 0 (hatched, Fig. 3(b)), assuming Type 1 interdependence and $S = 2$. At stage 2, every edge in both layers is touched by failed nodes, thus the initially failed nodes represent a vertex cover of this interdependent network (Fig. 3(a)). Fig. 4 illustrates the same situation assuming Type 2 interdependence. Due to a different nature of interdependence, the number of nodes in the vertex cover is three, which means that more nodes need to be disabled at stage 0 in order to achieve the same result by stage 2. Note that if both sets E_{12}, E_{21} are empty, then no failure propagation occurs, and the considered problem becomes the classical minimum vertex cover problem on a simple undirected graph $G1 \cup G2$.

4.3 Optimization Models

4.3.1 Type 1 Interdependence

A. IP formulation Denote $A^{11}, A^{22}, A^{12}, A^{21}$ as the adjacency matrices for the sets of edges E_1, E_2 and the sets of interdependent links E_{12}, E_{21}, respectively, with the corresponding elements $\alpha^{11}_{ij}, \alpha^{22}_{ij}, \alpha^{12}_{ij}, \alpha^{21}_{ij}$. Define the binary variables v^s_{li} that show whether node i in layer l $(l = 1, 2)$ fails by stage $s(v^s_{li} = 1)$ if it does fail, and $v^s_{li} = 0$ otherwise). Using these

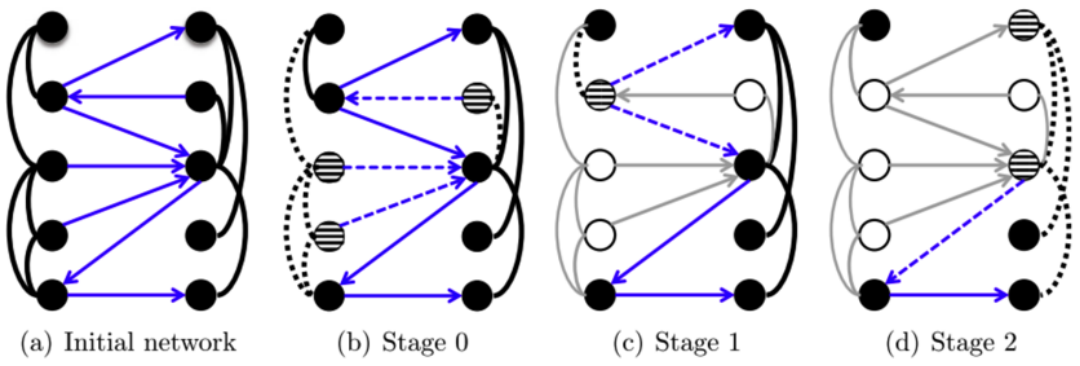


Figure 4.4: *Illustration of stage-by-stage node failures caused by Type 2 interdependence.*

variables, the linear 01 formulation for the minimum vertex cover in $G^{(2)}$ problem can be written as follows:

PROBLEM 1 (F1)

$$minimize \sum_{l=1,2}\sum_{i\in V_l} v_{li}^0 \tag{1}$$

$$subject\ to\ v_{li}^s + v_{lj}^s \geq 1, \forall (i,j) \in E_l, l = 1,2, \tag{2}$$

$$v_{1i}^s \geq \frac{1}{n_2+1}(v_{1i}^{s-1} + \sum_{j\in V_2} \alpha_{ji}^{21} v_{2j}^{s-1}), \forall i \in V_1, s = 1,...,S, \tag{3}$$

$$v_{2i}^s \geq \frac{1}{n_1+1}(v_{2i}^{s-1} + \sum_{j\in V_1} \alpha_{ji}^{12} v_{1j}^{s-1}), \forall i \in V_2, s = 1,...,S, \tag{4}$$

$$v_{1i}^s \leq (v_{1i}^{s-1} + \sum_{j\in V_2} \alpha_{ji}^{21} v_{2j}^{s-1}), \forall i \in V_1, s = 1,...,S, \tag{5}$$

$$v_{2i}^s \leq (v_{2i}^{s-1} + \sum_{j\in V_1} \alpha_{ji}^{12} v_{1j}^{s-1}), \forall i \in V_2, s = 1,...,S, \tag{6}$$

$$v_{li}^s \in \{0,1\} \forall i \in V_l, l = 1,2, s = 0,...,S, \tag{7}$$

Constraints (2) ensure that the nodes failed by stage S form a vertex cover in the entire two-layer network. Constraints (3) and (4) recursively model stage-by-stage node failures (in both layers) caused by the failures of nodes belonging to stage 0. Constraints (5) and (6) reflect the condition that the nodes that have not yet been affected by cascading failures by stage s remain operational at stage s.

The formulation (1)(7) contains $(S+1)n$ binary variables and $|E_1|+|E_2|+2Sn$ constraints. The next section describes a slightly more compact formulation for the same problem utilizing the nature of Type 1 interdependence, which contains n binary variables and $|E_1|+|E_2|$ constraints (i.e., the number of entities does not depend on S). Moreover, for the tighter LP relaxation, constraint (3) can have $\sum_{j\in V_2} \alpha_{ji}^{21}, \forall i \in V_1$ instead of n_2, and constraint (4) can have $\sum_{j\in V_1} \alpha^{12}_{ji}, \forall i \in V_2$ instead of n_1. Although we use these observations in the computational experiments, for the sake of readability we continue further discussion using the normalizing coefficients in the above format.

B. More compact IP formulation

For any node i in layer $m(m = 1, 2)$ define $D^{rm}(i)$ as the set of nodes in layer $r(r = 1, 2)$ such that failure of any node in $D^{rm}(i)$ at stage 0 causes node i to fail by stage S, which can be formally written as $D^{rm}(i) = \{k \in V_r :$ if k fails at stage 0, then $i \in V_m$ fails by stage S$\}$.

To simplify further notation, define $D^{rm}(i, j) = D^{rm}(i) \cup D^{rm}(j)$. We will use the notation $D^{rm}(i, j, S)$, or $D^{rm}(i, S)$, if we need to emphasize its dependency on the number of considered cascade stages S. Consequently, all affected nodes at stage S form a vertex cover if and only if for any edge (i, j) in layer 1 there is at least one failed (attacked) node in the set $D^{11}(i, j)\cup D^{21}(i, j)$, and for any edge (i, j) in layer 2 there is at least one failed (attacked) node in the set $D^{22}(i, j) \cup D^{12}(i, j)$. Using this observation and the definition of $D^{rm}(i)$, the problem can be formulated as follows:

PROBLEM 1 (F1c)

$$minimize \sum_{l=1,2} \sum_{i\in V_l} v_{li} \tag{9}$$

$$subject\ to: \sum_{k\in D^{11}(i,j)} v_{1k} + \sum_{k\in D^{21}(i,j)} v_{2k} \geq 1, \forall (i, j) \in E_1 \tag{10}$$

$$\sum_{k\in D^{22}(i,j)} v_{2k} + \sum_{k\in D^{12}(i,j)} v_{1k} \geq 1, \forall (i, j) \in E_2 \tag{11}$$

$$v_{ij} \in \{0, 1\}, \forall i \in V_l, l = 1, 2.$$

This problem formulation requires the identification of sets $D^{rm}(i)$ for every node i in the network. This can be easily done using the reverse breadth-first search from i up to a level S since $D^{rm}(i)$ is the set of nodes in V_r from which there is a shortest directed path of length $\leq S$ to node $i \in V_m$ through E_{12}, E_{21}.

C. LP relaxation approximation

It is a well-known fact that an LP relaxation of the minimum vertex cover problem provides a 2-approximation algorithm (Vazirani, 2001). In this section we analyze an approximation algorithm for the minimum vertex cover problem in interdependent networks based on the LP relaxation of formulation (9)(11).

Proposition 1
Define Δ_1 and Δ_2 as

$$\Delta_1 = \max_{(i,j)\in E_1} |D^{11}(i, j)| + |D^{21}(i, j)|,$$

$$\Delta_2 = \max_{(i,j)\in E_2} |D^{22}(i,j)| + |D^{12}(i,j)|,$$

and $\Delta = max(\Delta_1, \Delta_2)$. Then, the LP relaxation of formulation (9)-(11) provides a Δ-approximation.

Proof. Let $v^*_{li}, i = 1, ..., n_l, l = 1, 2$ be an optimal solution of the LP relaxation of the IP formulation (9)(11), and let w_{LP}, w_{IP} be the optimal objective values of LP and IP, respectively. Obviously, $w_{LP} \leq w_{IP}$. Define for $l = 1, 2$

$$v_{li} = \begin{cases} 0 & \text{if } v^*_{li} < 1/\Delta_l, \\ 1 & \text{if } v^*_{li} \geq 1/\Delta_l. \end{cases}$$

Since the number of variables in every constraint of the formulation (9)(11) is no greater than Δ, then vector $v_l i, i = 1, ..., n_l, l = 1, 2$ is a feasible solution of the IP (9)(11). Let w^{LP}_{IP} be the corresponding IP objective function of this solution, then $w_{IP} \leq w^{LP}_{IP}$.

Also, by definition of v_{li} it follows that $w^{LP}_{IP} \leq \Delta w_{LP}$. Combining the last two inequalities, we get $w_{IP} \leq \Delta w_{LP}$.

□

Note that if there are no interdependent links between the layers ($E_{12} = \emptyset, E_{21} = \emptyset$), then $\Delta = 2$, and the problem (9) - (11) reduces to the standard minimum vertex cover problem. Therefore, Proposition 1 contains the classical 2-approximation result for the standard minimum vertex cover problem as a special case.

D. Depth of cascade S^*

As mentioned above, the total number of stages in a cascading failure process cannot exceed $S = n-1$ (the longest possible cascade), since at every stage the cascade either stops or causes at least one more node failure. Therefore, if one wants to solve the considered problem with unconstrained number of failure stages, then $S = n - 1$ should be used in the aforementioned problem formulations ($F1, F1c$). However, for computational reasons, it might not be efficient to use the largest possible S, since it may require a lot of unnecessary calculations and usage of redundant variables. We show how to find such a number of failure stages $S^* \leq n-1$, that the problem formulations $F1(S)$ and $F1c(S)$ have the same optimal solution for all $S \geq S^*$. In the notations $F1(S)$ and $F1c(S)$, we use the argument S to refer to the problem formulations $F1, F1c$ with a specified parameter S.

Definition 5: The **depth of cascade** S^* is the maximum possible length of the sequence of cascading failures in a given interdependent network $G^{(2)}$. The next proposition shows that in the case of Type 1 interdependence, the depth of cascade can be identified explicitly for any given problem instance.

Proposition 2: Let $G^{(2)} = ((V_1, E_1), (V_2, E_2), E_{12}, E_{21})$ be an interdependent network with Type 1 interdependence, $G = (V_1 \cup V_2, E_{12} \cup E_{21})$, and let $dist(i, j)$ be the length of the shortest directed path from node i to node j in G ($dist(i, j) = \infty$ if there is no path from i to j). Then, the **depth of cascade** of network $G^{(2)}$ is given by $S^* =$

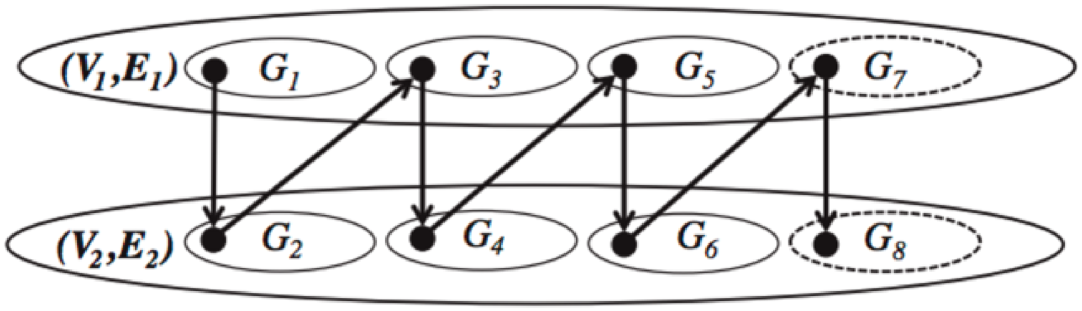


Figure 4.5: *Illustration of a constructed interdependent network for $S = 5, S^* = 7$ in the proof of Proposition 3. Dashed objects contain only isolated nodes.*

$\max_{i,j \in V_1 \cup V_2}(dist(i,j)|dist(i,j) < \infty)$ Moreover, formulation $F1c(S)$ are identical for all $S \geq S^*$.

Proof. The proof is straightforward and follows from the definition of S^*.

□

4.3.2 Type 2 Interdependence

Similarly to Type 1 interdependence, one can consider the concept of failure stages $s = 0, ..., S$ introduced previously in the context of Type 2 interdependence. The corresponding linear 01 problem formulation is obtained using the same notations as in formulation (1)(7) by changing the constraints on stage-by-stage node failures to incorporate this type of interdependence.

$$minimize \sum_{l=1,2} \sum_{i \in V_l} v_{li}^0 \tag{12}$$

$$subject\ to\ v_{li}^s + v_{lj}^s \geq 1, \forall (i,j) \in E_l, l = 1,2, \tag{13}$$

$$v_{li}^s \geq v_{li}^{s-1}, \forall i \in V_l, l = 1,2, s = 1, ..., S, \tag{14}$$

$$1 - v_{1i}^s \geq \frac{\sum_{j \in V_2} \alpha_{ji}^{21}(1 - v_{2i}^{s-1})}{n_2 + 1} - v_{1i}^{s-1}, \forall i \in V_1 : \sum_{j \in V_2} \alpha_{ji}^{21} \geq 1, s \leq S, \tag{15}$$

$$1 - v_{2i}^s \geq \frac{\sum_{j \in V_1} \alpha_{ji}^{12}(1 - v_{1i}^{s-1})}{n_1 + 1} - v_{2i}^{s-1}, \forall i \in V_2 : \sum_{j \in V_1} \alpha_{ji}^{12} \geq 1, s \leq S, \tag{16}$$

$$1 - v_{1i}^s \leq \sum_{j \in V_2} \alpha_{ji}^{21}(1 - v_{2i}^{s-1}), \forall i \in V_1 : \sum_{j \in V_2} \alpha_{ji}^{21} \geq 1, s \leq S, \tag{17}$$

$$1 - v_{2i}^s \leq \sum_{j \in V_1} \alpha_{ji}^{12}(1 - v_{1i}^{s-1}), \forall i \in V_2 : \sum_{j \in V_1} \alpha_{ji}^{12} \geq 1, s \leq S, \tag{18}$$

$$v_{li}^s \in \{0,1\}, \forall i \in V_l, l = 1,2, s = 0, ..., S.$$

Note that the theoretical results proven for the problem with Type 1 interdependence do not necessarily extend automatically for Type 2 interdependence due to the substantially different structure of failure mechanisms, which makes it challenging to address from analytical and computational perspectives.

4.3.3 Mixed Interdependence

In addition to the two aforementioned baseline models, where the same type of interdependence (e.g., Type 1 or Type 2) is characteristic for both network layers (and every individual node in these layers), it is easy to extend the minimum vertex cover problem to a more general case, where network layers are characterized by different types of interdependence, for instance, Type 1 interdependence for layer $G_1 = (V_1, E_1)$ on layer $G_2 = (V_2, E_2)$, and Type 2 interdependence for layer G_2 on layer G_1. In this case, the linear 01 programming formulation is obtained straightforwardly from formulation F2 by substituting constraints (15) and (17), which correspond to the failure of nodes in layer $G_1 = (V_1, E_1)$ due to Type 2 dependence on $G_2 = (V_2, E_2)$, by constraints (3) and (5) which correspond to the failure of nodes in layer $G_1 = (V_1, E_1)$ due to Type 1 dependence on $G_2 = (V_2, E_2)$. Moreover, the formulation can be generalized similarly if every node in the interdependent network has its own type of interdependence (Type 1 or Type 2) on nodes from the other layer.

4.4 Computational Complexity

The classical Minimum Vertex Cover (MVC) problem, which is NP-hard (Garey & Johnson, 1979), is a special case of the Minimum Vertex Cover problem in Interdependent Networks (MVCIN) for $S = 0$. However, it does not necessarily imply that the MVCIN problem is NP-hard for any fixed S and S^*. The propositions below show that the decision (recognition) version of MVCIN problem is NP-complete for any fixed S, S^*. Following the standard approach (Garey & Johnson, 1979), we define the recognition version of the problem, Vertex Cover in Interdependent Networks (VCIN), as follows: given fixed positive integers $k, S, S^* (S \leq S^*)$ and a two-layer interdependent network $G^{(2)}$ with depth of cascade S^*, does there exist a subset of nodes $V' \subseteq V_1 \cup V_2$ of size k whose failure causes the failure of a subset of nodes $V'_c \subseteq V_1 \cup V_2$, which forms a vertex cover in $G^{(2)}$?

Proposition 3. The VCIN problem with Type 1 interdependence is NP-complete for any fixed S, S^*.

Proof. The VCIN problem is obviously in class NP. Next, we reduce the classical VC problem to the VCIN problem. Namely, for the given k, S, S^* and a graph $G = (V, E)(V = \{1, ..., n\})$ we will construct a two-layer interdependent network $G^{(2)} = ((V_1, E_1), (V_2, E_2), E_{12}, E_{21})$ with depth of cascade S^*, and prove that G has a vertex cover of size k if and only if $G^{(2)}$ has a vertex cover of size k for the fixed number of failure stages S. The construction proceeds as follows. Let

$$G_i = \begin{cases} (V, E) & i = 1, ..., S + 1, \\ (V, \emptyset) & i = S + 2, ...S^* + 1. \end{cases}$$

In other words, G_i is just a copy of graph G if $i \leq S+1$, and is a graph with $|V|$ nodes and no edges if $S+1 < i \leq S^*+1$. To distinguish the nodes among the graphs G_i we denote a node $j \in V$ in G_i as j_i. Let also

$$(V_1, E_1) = \bigcup_{i=1}^{\left\lceil \frac{S^*+1}{2} \right\rceil} G_{2i-1}, (V_2, E_2) = \bigcup_{i=1}^{\left\lceil \frac{S^*+1}{2} \right\rceil} G_{2i}$$
$$E_{12} = \left\{ (j_{2i-1}, j_{2i}) : \forall j \in V; i = 1, ..., \left\lceil \tfrac{S^*+1}{2} \right\rceil \right\}$$

$$E_{21} = \left\{ (j_{2i}, j_{2i-1}) : \forall j \in V; i = 1, ..., \left\lceil \tfrac{S^*-1}{2} \right\rceil \right\}$$

The depth of cascade of this interdependent network is, obviously, S^*, and the network construction can be done in a polynomial time.

Fig. 5 illustrates how the interdependent network $G^{(2)}$ is constructed from $G = (V, E)$ for $S = 5, S^* = 7$. Observe that the failure of node j_1 in G_1 will consequently cause the failure of nodes j_i in all graphs G_i for $i = 2, ..., S+1$. Assume that $V' \subset V$ is a vertex cover of graph G, such that $|V'| = k$. Let $V^{(2)} = i_1 \in V_1 : \forall i \in V'$, then, obviously, $V^{(2)}$ is a vertex cover of the interdependent network $G^{(2)}$. Conversely, assume that $V^{(2)}$ is a vertex cover of size k in the interdependent network $G^{(2)}$. Using the observation above, we can conclude that having a node $j_i \in V^{(2)} (i > 1)$ does not do better than having a node $j_1 \in V^{(2)}$. Indeed, the failure of node j_1 will cause the failure of all the nodes that fail due to the failure of node j_i for any fixed $i > 1$. Therefore, without loss of generality, we can assume that all the nodes in $V^{(2)}$ belong to the first copy of graph $G(G_1)$. Obviously, the set of nodes $V^{(2)}$ in the original graph G is its vertex cover of size k. □

Proposition 4. The VCIN problem with Type 2 interdependence is NP-complete for any fixed S, S^* .

Proof. Using the proof of Proposition 3, observe that in the constructed interdependent network $G^{(2)}$ Type 2 interdependence produces the same failure mechanism, since every node in $G^{(2)}$ depends on no more than one other node. Therefore, the same logic can be used to prove the NP-completeness of VCIN problem with Type 2 interdependence. □

Remark 1. The NP-completeness results are straightforwardly generalized to the VCIN problem with mixed types of interdependence.

4.5 Adding Node Capacities and Loads

4.5.1 General Dependencies

In this case, we extend our model by adding general dependencies. Each node $i \in V_1$ and each node $j \in V_2$ has also an integer number d_i, d_j respectively, describing the number of active nodes in the opposite layer that a node needs to be secured/functioning in order for the node to function. If at one stage, the number d_i, for example, of node i is greater than the number of functioning nodes (in the opposite layer) it depends on, then node i will fail. We can say, that the number d_i is the least number needed for i to operate. The same as above hold for nodes in V_2. This kind of model may be useful in scenarios where a node is

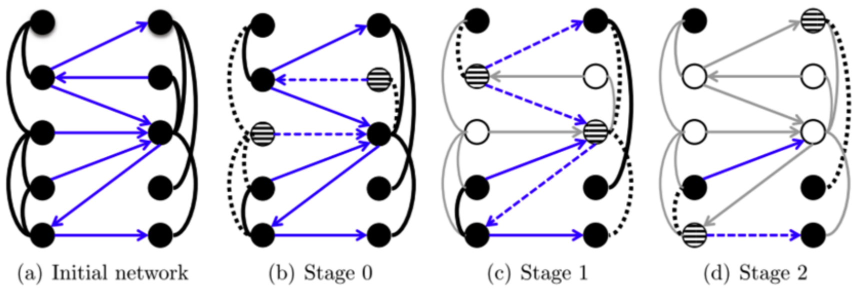


Figure 4.6: *In this case (Type 1) we have dependencies (0,1,0,0,1) for the set V_1 and dependencies (1,0,3,0,1) for the set V_2*

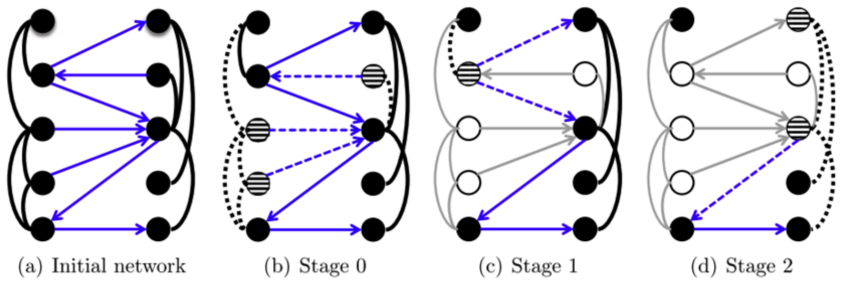


Figure 4.7: *In this case (Type 2) we have dependencies (0,1,0,0,1) for the set V_1 and dependencies (1,0,1,0,1) for the set V_2*

connected and receives some kind of resources (eg. electricity) by many other nodes, but not all of them are necessary for its functionality; it rather needs a minimum number of active nodes in order to function.

We give in Figures 6,7 some previous examples with the extra parameters added.

Proposition 5. The problem VCIN-GD (general dependencies) as described above is NP-complete.

Proof. The proof is easy since this problem with general dependencies is a generalisation of VCIN with Type 1 or with Type 2 dependencies (and the problem with "mixed dependencies" as well): we can fix the dependencies of each node (that is the numbers d_i, d_j) so that we take instances of Type 1 or Type 2 dependencies. For the Type 1 interdependence all we have to do is set d_i to the number of dependencies that node i has; that is the number of in-going edges $(., i)$ in the input set E_{21}. The same hold for nodes $j \in V_2$. Then if some node fails then immediately all dependent nodes will fail modelling the Type 1 dependence. For the Type 2 interdependence all we have to do is set $d_i = 1$ if i has any dependencies and $d_i = 0$ if it is completely independent. The same hold for nodes $j \in V_2$; we set $d_j = 1$ or 0, accordingly. Then, for some node to fail it must be the case that all the nodes it depends on have already failed, modelling the Type 2 dependence. The case of mixed interdependence can be modelled as well. □

4.5.2 Load Redistribution and Capacity Failures

Interdependent systems may as well be characterised by loads and capacities. The nodes in one system may be servers responsible for handling queries and the other system might be the power system providing electricity. Thus, it might be the case that we have load redistribution from failed nodes towards other active nodes on top of the interdependencies. The load redistribution is between nodes of the same layer (you cannot have load redistribution from a PC to a power system) and it depends on the intra-edges of each network. The nodes also have capacities denoting the maximum load they can handle. Thus, failures may arise not only because of the interdependencies but also due to excessive loads in specific nodes that receive too much load due to load redistribution. In order to capture this aspect of real life scenarios, we must extend our initial model (not with the general dependencies exactly above) by associating with each node i two numbers: its load L_i and its capacity C_i. These numbers are given for all nodes in each layer as an input. A node is functioning, if firstly the dependencies it needs from the other layer are all satisfied (that is the nodes it depends on are functioning normally) and secondly, it holds true that its load is not exceeding its capacity $L_i \leq C_i$. A node fails, in other words, either if some of the nodes it depends on fail, or if its loads becomes too much and exceeds its capacity.

Proposition 6. The problem VCIN-LC (with loads and capacities) as described above is NP-complete.

Proof. This problem is a generalisation of the VCIN problem that we proved to NP-complete in the previous section. To see this we can set all capacities to infinity ($C_i = \infty$, or set them to the sum of all loads). Then, no cascading failures will happen between the two layers due to load redistribution and the only failures will be because of the interdependencies. □

Finally, we can combine the two problems of this section, the VCIN-GD and the VCIN-LC and obtain the most general version of the vertex cover problems in interdependent networks which is of course NP-complete.

4.6 Conclusion

In this chapter, we have introduced an optimization-based framework for modeling cascading failures in interdependent networks, as well as the impact of two common types of interdependence on network robustness characteristics. As a quantitative measure of network robustness, the optimal solution of the minimum vertex cover problem was utilized; however, the setup of this classical optimization problem in interdependent networks needed to be extended due to the difference in the nature of such networks compared to standard single-layer networks. Our results show that this extended modeling approach can incorporate different types of interdependence that are common in practice; moreover, certain properties of the classical vertex cover problem (i.e., the computational complexity and the LP approximation ratio) can be generalized for the setup with interdependent networks. We also studied the case with general number of dependencies and the case where the nodes are characterised by loads and capacities under load redistribution of the failed nodes, and proved hardness results in both cases. Possible extensions of the proposed approach include the following directions. First, the introduced formulations may allow one to incorporate and optimize other measures of network robustness (besides the size of the minimum vertex cover) by utilizing a different objective function and possibly an extra set of constraints in addition to the ones modeling two types of interdependence. Second, this framework may be extended to interdependent networks with more than two layers. Third, the proposed formulations can be generalized by incorporating edge and/or node weights. These weights can represent, for instance, capacity parameters, costs of disabling and/or protecting a node, as well as probabilities of node failures due to dependencies on other nodes (i.e., failures occurring probabilistically rather than deterministically). In the latter setup, quantitative measures of risks associated with cascading failures may be incorporated in the corresponding optimization models.

Chapter 5

System Fragility under Load Redistribution from an Attacker's Perspective

5.1 Cascading Failures Dynamics

In this section, we will use the $L - C$ model: each node has 2 numbers characterising it and these are its load L and its capacity C. We describe how the failures propagate in our $L - C$ model when we attack a node or multiple nodes.

What does equal redistribution mean and what are the dynamics of the cascading process?

To be clear, we say that we have topology knowledge if we assume the network to have some kind of structure in the sense that not all nodes/advertisers are the same. When the graph that represents the network is irrelevant to our goals and to our problems' statements we say that there is no topology knowledge. Having this in mind, equal redistribution refers to the fact that the load of the failed node is democratically shared among the remaining well-functioning nodes. We make a distinction as it will be obvious below in two main cases: where we have no topology knowledge and where we do have topology knowledge. In the former case, the load is redistributed to all other well-functioning nodes, whereas in the latter case it is redistributed towards only the well-functioning neighbours of the failed node. To make the above statement precise, we would like to give an example:

EXAMPLE: When we attack an advertiser, let's say the advertiser (L_0, C_0), and we are in the case of no topology knowledge, then his load needs to be redistributed equally to the rest $n-1$ advertisers. This means that a portion $\dfrac{L_0}{n-1}$ will be added to each other advertiser's load. So the $i-th$ advertiser will now have $L_i + \dfrac{L_0}{n-1}$ as his load. If this quantity is strictly **less** to his capacity C_i then this advertiser survives. However, if for some advertiser, let's say advertiser 1, the quantity $L_1 + \dfrac{L_0}{n-1} \geq C_1$ then the advertiser 1 will fail as well and then we will have to distribute both $L_0 + L_1$ to $n-2$ advertisers and so on, until no more cascading failures happen. We say that we reached the end of the cascading failures. To point out the difference, if we had topology knowledge then the load of the attacked advertiser A_0, L_0, would be shared only among all his out-neighbours. Because of the extra load, each node $i \in N(A_0)$ (out-neighbourhood of A_0) will have total load $L_i + \frac{L_0}{|N(A_0)|}$. Again, if this quantity is strictly **less** to his capacity C_i then this advertiser survives. However, if for some neighbour, let's say advertiser A_1, the quantity $L_1 + \frac{L_0}{|N(A_0)|} \geq C_1$ then the advertiser A_1 will fail as well and then we will have to distribute both $L_0 + L_1$ to $|N(A_0 \cup A_1)|$ advertisers and so on, until no more cascading failures happen. We say that we reached the end of the cascading failures. There is an important fact to observe in our redistribution model, in the latter case, if our attack consists of multiple nodes. If these nodes form a weakly connected component consisting only of failed nodes, i.e. there is a path (ignoring the directions) from any attacked node to any other passing through failed nodes only, then we can redistribute their sum of loads as described above, i.e. to the out-neighbourhood of $\cup_{i \in attacked} A_i$. But if they do not form a weakly connected component, then the redistribution is happening separately to each neighbourhood of every attacked node, proceeding independently until the cascades stop or the failures have an overlap. This scenario is depicted in Figure 3.

The following two versions of Load Redistribution problems are important for our analysis. The first version supposes no topology knowledge whereas in the second we take into account how the network might behave in case of failures on the nodes.

5.2 ER-k-min $\sum L$ Problem

The **Equal Redistribution** Problem with k attacks and minimum $\sum L$ is the problem in which we are interested in attacking a set of advertisers/containers A, of size k, having minimum total load $\sum_{i \in A} L_i$ so that the whole system will be destroyed; that is after the cascading failures no advertiser is left functioning. The objective is twofold and we have a tradeoff: by attacking "heavier" containers (bigger loads) we have better chances to destroy the whole system due to cascading failures, however, we are aiming at minimizing the total load of the attacked containers. This knapsack-like tradeoff is what makes the problem NP-complete as we will prove in the next section.

More formally stated the optimisation problem is the following :

INPUT: A list of n pairs of non-negative rational numbers in the form (L_i, C_i) indicating the load and the capacity of each advertiser, an integer $0 < k \leq n$ and an integer $0 < k' \leq n$. We suppose $C_i > L_i$ that is no advertiser is failing initially at its own load.

OUTPUT: An attack set A of advertisers, with $|A| = k$ and minimum total load $\sum_{i \in A} L_i$, so that at the end of the cascading failures the number of failed nodes is at least k'.

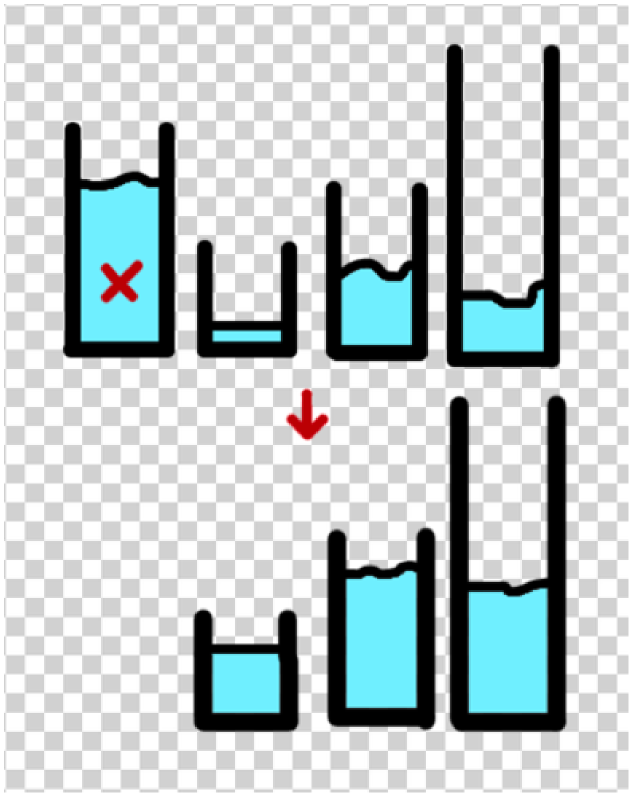


Figure 5.1: *Illustrative example for the case of no topology knowledge. The first container having a load of 9, after the attack, redistributed its load to the three alive containers by giving an extra load of 3 to each. No further cascading failures took place.*

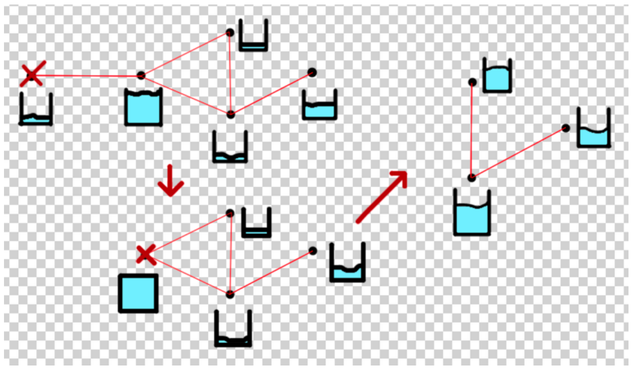


Figure 5.2: *Illustrative example for the case we have the topology knowledge. The first attacked advertiser* A_0 *had a load of 1 and he gave all of his load to his single neighbour* A_1 *with* $L = 4, C = 5$ *who was on the verge of destruction. After his destruction, the total load of* $1 + 4 = 5$ *is redistributed to the two neighbours of* $A_0 \cup A_1$ *(extra load of 2,5 to each of the neighbours). They remain alive and the system reaches the end of cascades.*

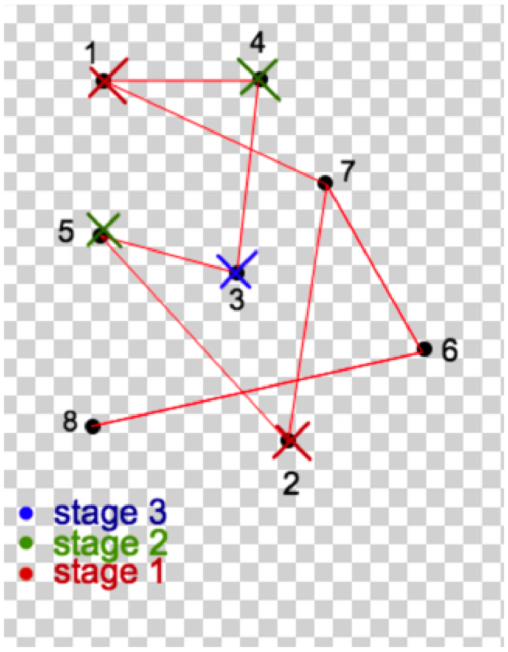


Figure 5.3: *Illustrative example for multiple attacks in the case we have topology knowledge. The instance is consisting of 8 nodes namely 1,2,3,4,5,6,7,8 with loads and capacities respectively: (2,3), (2,3), (0,2), (0,1), (0,1), (1,7), (1,7), (1,7). The first stage is the attack to nodes 1,2. Since they do not form a weakly connected component with failed nodes only, the load of 1 is equally redistributed to 4 and 7; the load of 2 is equally redistributed to 5 and 7. Nodes 4 and 5 will fail (stage 2) since their load will be equal to their capacities* ($L_4 = L_5 = 1 = C_4 = C_5$)*. Now we must take nodes 1,4 as a union and also nodes 2,5 as a union and redistribute their total load to their alive neighbours 3,7. Node 7 will survive but node 3 will also fail (stage 3). Now the failed nodes 1,2,3,4,5 are a connected component and we take them as a union and redistribute their total load to node 7. The final situation is: X,X,X,X,X,(1,7),(5,7),(1,7), where X means failed.*

More formally stated the decision problem is the following :

INPUT/OUTPUT: A list of n pairs of non-negative rational numbers in the form (L_i, C_i) indicating the load and the capacity of each advertiser, an integer $0 < k \leq n$, an integer $0 < k' \leq n$ and a rational number L. We suppose $C_i > L_i$ that is no advertiser is failing initially at its own load. We are interested in knowing whether or not there is an attack set A with size exactly k, and total sum of loads $\sum_{i \in A} L_i \leq L$, so that at the end of the cascading failures the number of failed nodes is at least k'.

5.3 ERAN-k Problem

The **Equal Redistribution Among Neighbours** Problem with k attacks is the following: We are given a network representing the potential flow of load (direction of load redistribution) in case of node failures. So the redistribution is among the neighbours of the failed node and it is supposed to be done equally. Then, we are asked to identify k nodes to attack so that we maximize the number of failures.

More formally stated the optimisation problem is the following :

INPUT: A directed graph $G = (V, E)$ where $|V| = n$ and a list of n pairs of non-negative rational numbers in the form (L_i, C_i) indicating the load and the capacity of each node/advertiser and an integer $0 < k \leq n$. We suppose $C_i > L_i$ that is no advertiser is failing initially at its own load.

OUTPUT: An attack set A of advertisers, with $|A| \leq k$ so that at the end of the cascading failures the number of failed nodes is maximized.

More formally stated the decision problem is the following :

INPUT/OUTPUT: A directed graph $G = (V, E)$ where $|V| = n$ and a list of n pairs of non-negative rational numbers in the form (L_i, C_i) indicating the load and the capacity of each node/advertiser, an integer $0 < k \leq n$ and another integer $0 < k' \leq n$. We suppose $C_i > L_i$ that is no advertiser is failing initially at its own load. We are interested in knowing whether or not there is an attack set A with size $|A| \leq k$, so that at the end of the cascading failures the number of failed nodes is at least k'.

5.4 ERAN-k & Greedy Algorithms that Fail

Here we will present 3 intuitive greedy algorithms and give tight examples proving their bad performance for the optimisation ERAN-k Problem in the very special case where the topology is the complete undirected clique K_n, where n are the number of nodes and we have $k' = n$, i.e. we want to destroy the whole system. We give a more complete example about the cascading failures and the total collapse of the system. We can again view the procedure as a fun game with water and containers:

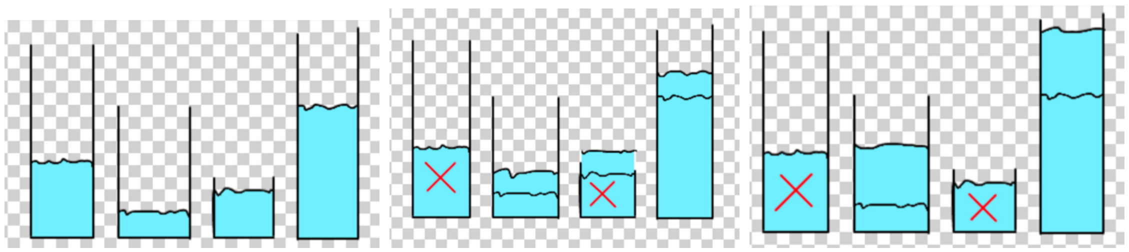


Figure 5.4: *For example this instance is the (6,16) (2,11) (4,5) (11,17). We choose to attack the first container and then we have* $\frac{6}{3} = 2$ *units of water to pour to the other 3 containers. The third should have* $4 + 2 = 6 > 5$ *units and thus will be destroyed. Now the situation is like having the content of the first and third container (6+4=10 units of water) and we want to pour it into the second and the fourth container. So* $\frac{10}{2} = 5$ *extra units of water in each. In the final situation we have no cascading failures and the survived containers have (7,11),(16,17). The optimal solution is to attack the fourth along with the first or second container.*

In Figure 4, we see 4 containers that each have different capacities and different loads. We want to choose which is the minimum number of containers to attack in order to destroy all of them. If we attack to the first container then there are 3 containers left so we divide the first's load to 3 parts and we add each part to the load of the other containers. The third container will fail due to low capacity and we must pour it as well to the two others, the second and the fourth. The intermediate phases as well as the final situation where the cascading failures have stopped is depicted in Figure 4.

One main observation is that the following greedy algorithms do not work and actually can deviate a lot from the optimal solution :
*A)*Attacking the container with greatest load: This strategy aims to maximize the load that we will redistribute in each attack round by maximizing the nominator L_0 of $\frac{L_0}{n-1}$. This strategy fails because there may be containers with pretty big capacities and this greedy approach doesn't take these capacities into account. The deviation from the optimal can be $\Theta(n)$. (the attacked are n versus the optimal which is 1) (Figure 5)

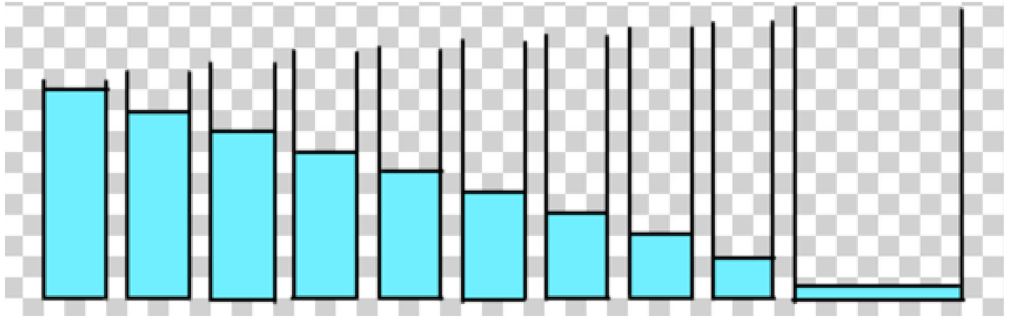


Figure 5.5: *In this counterexample (to the max load attack) we have:* $(10, 10 + \frac{1}{10}), (9, 9 + \frac{10}{9} + \epsilon), (8, 8 + \frac{19}{8} + \epsilon), (7, 7 + \frac{27}{7} + \epsilon), (6, 6 + \frac{34}{6} + \epsilon), (5, 5 + \frac{40}{5} + \epsilon), (4, 4 + \frac{45}{4} + \epsilon), (3, 3 + \frac{49}{3} + \epsilon), (2, \frac{52}{2} + \epsilon), (1, 1 + \frac{54}{1} + \epsilon)$ *where* $\epsilon > 0$ *is really small compared to any of the loads. The greedy will output* $k = 10$ *since it will start attacking the first/leftmost container with load* $L_1 = 10$ *and no cascading failures will happen and then it will continue towards the right. The optimal solution is* $k = 1$ *by attacking the last container because then the cascading failures will take place (the first container will then fail since it will have* $L_1 > C_1$*), thus destroying the whole system. Of course, we can generalise the above counterexample and instead of* 10 *we could have* n *containers with the greedy output being* $k = n$ *whereas the optimal solution being* $k = 1$

*B)*Attacking the container with greatest capacity: This strategy in each round attacks the container that has the maximum capacity since this might take out large containers and so the remaining, supposedly small containers will be destroyed due to load redistribution. This strategy fails to achieve the optimal as well, because there may be containers with pretty big capacities but small enough loads (or even almost empty) that cannot destroy other containers, and this greedy approach doesn't take these cases into account. The deviation from the optimal can be $\Theta(n)$. (the attacked are $n+1$ out of $2n+1$ versus the optimal which is 1) (Figure 6)

*C)*Attacking the container with greatest (*capacity* − *load*) difference: This approach wants to take advantage of the fact that when a container's load and capacity are close, then there is no need to attack it since it will fail soon enough due to load redistribution. This greedy strategy also fails because the relative magnitudes of the loads and capacities of the containers matter a lot and this fact is not taken into account. For example, one container may have large (*capacity* − *load*) difference but it may be the case that its load is negligible comparably to the other containers' capacities. The deviation from the optimal can be $\Theta(n)$. (attacked: n versus optimal: 1) (Figure 7)

5.4.1 Attack Projection Algorithms

Three main observations that are important for an algorithmic approach to the equal redistribution scheme are the following :

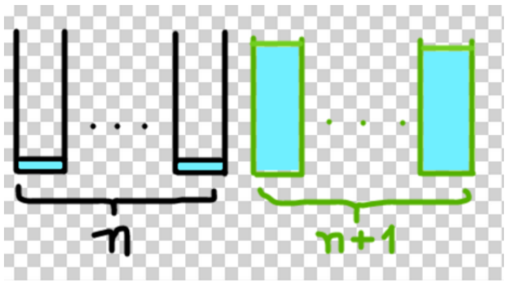


Figure 5.6: *In this counterexample (to the max capacity attack) we have $2n+1$ containers with (ϵ, M) for the first n containers (the big containers) and $(M-2\epsilon, M-\epsilon)$ for the last $n+1$ containers (the small but almost full containers). The greedy will output $k = n+1$ since it will start attacking the first/leftmost n containers first with capacity $C_i = M$ and no cascading failures will happen and then it will continue towards the right. When the first small but almost full container fails then everything fails and the system breaks down.The optimal solution is $k = 1$ by attacking the last container because then the cascading failures will take place destroying the small containers along with the big but empty containers.*

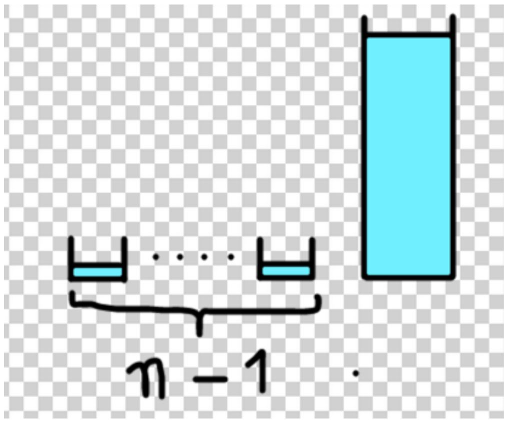


Figure 5.7: *In this counterexample (to the max (capacity−load) attack) we have n containers with $(\epsilon, (n+1)\epsilon)$ for the first $n-1$ containers (the small containers thought with large $C-L$ difference) and $(M, M+(n-1)\epsilon)$ for the last container (the big container relative to the others though with smaller $C-L$ difference), where M is chosen to be such that $M > (n^2-1)\epsilon$ is true. The greedy will output $k = n$ since it will start attacking the leftmost containers first and no cascading failures will happen and then it will continue towards the right. The algorithm must destroy all the containers in order for the system to break down.The optimal solution is $k = 1$ by attacking the last container because then the cascading failures will take place destroying the system.*

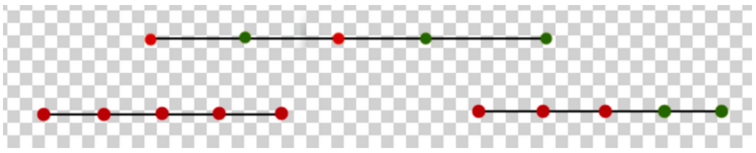


Figure 5.8: The order we choose to make the attacks in the case of ERAN-k is important. The instance's loads and capacities are (1,4),(1,2),(2,5),(1,3),(1,3). In the first chain above, we will attack the 2 red nodes. If we choose to attack starting from the left the result is different than if we start the attacks from the right.

A) Timeline of failures as the cascading effect takes place: If we sort the containers by increasing $fp = (C_i - L_i)$ difference then surely the first container will be the first to be destroyed due to load redistribution and so on; generally this will be the order by which the containers will get destroyed. (we note that from this sequence we exclude the containers that we are attacking to)

B) Sufficient (but not necessary) condition to destroy the system: We denote with the symbol $Cont$ the set of containers and with the symbol A the containers that are under attack, that is, the containers our algorithm chooses at the beginning to destroy. We observe that a sufficient condition to achieve our goal of destroying the whole system is the following: $\sum_{i \in A} L_i \geq \sum_{i \in Cont \setminus A}(C_i - L_i)$. This means that the total load of the containers we are currently attacking exceeds the total of the empty space of the remaining containers. Of course, this condition is not necessary since it might be the case that we have extensive cascading failures, so that even by starting from a small container attack, ultimately more and more containers will get destroyed. This thought leads us to a broader condition. First, let us define the attack projection of a set of containers A as $proj(A)$ (something similar to the influence of set A) as the set of the attacked containers plus(union) the containers that will be destroyed after all cascading failures have stopped. We are interested in finding an initial set A to attack, we call it seed set A, with the goal to have $proj(A) = Cont$; then, all containers will have been destroyed and the system will collapse. The attack projection is calculated step by step by the algorithm given below and the broader condition is then: If at some point, starting with attack seed set A we have $\sum_{i \in proj(A)} L_i \geq \sum_{i \in Cont \setminus proj(A)}(C_i - L_i)$ then the system is guaranteed to break down.

C) The order of our attacks: In the equal redistribution scheme the order by which we decide to make our attacks will not affect the final outcome. This is because the load of the attacked nodes will be distributed to all of the remaining nodes so at the end an amount of $\sum_{i \in A} L_i$ will end up in the remaining containers (leading to new failures or not) no matter what the order we chose to actually attack the containers in the attack set A. The situation is different when we deal with the ERAN-k problem and the topology is known, because an attack might result in having the system's graph disconnected inhibiting the flow of load to other nodes and thus cancelling the cascading failures effect as we will see afterwards. A simple example is given in Figure 5.8.

In what follows we give two algorithms, the ER-Attack Projection algorithm that finds the impact an attack set A has and the Max-X (X=Load or Capacity) Greedy Algorithm for Special Cases that attacks the container with maximum X value first (X is either load or capacity) and that we will use in some special cases defined below.

Algorithm 1 ER-Attack Projection *proj(A)*

```
Require: Input is sorted according to FreeSpace: $fs = L_i - C_i$ for the $binarySearch$
Require: $binarySearch$ returns all containers with free space less than extra_load
 1: procedure $proj$(A)
 2:     new_set= $A$
 3:     repeat
 4:         previous _set = new_set
 5:         extra_load = $\sum_{i \in previous\_set} L_i$ / $(n - |previous\_set|)$
 6:         new_set= $binarySearch$(extra_load)$\cup$previous_set
 7:     until new_set = previous_set
 8:     Return new_set
 9: end procedure
```

Algorithm 2 Max-X (X=Load or Capacity) Greedy for Special Cases

```
Require: Input is sorted according to Loads
 1: procedure $max$-$X$ GREEDY(k)
 2:     repeat
 3:         Attack=$arg\mathbf{max}(X_i)$
 4:         Failed=$proj$(attack)$\cup$Failed
 5:         Input=Input\Failed                                ▷ we discard the failures
 6:         $k = k - 1$
 7:     until $k = 0$ or |failed|= $n$
 8: end procedure
```

5.5 ER-k & Greedy Algorithms that Succeed

The ER-k problem has many variations depending on the relations between the Loads L_i and the Capacities C_i. In the following, the words advertisers, nodes, stores might be used interchangeably and indistinguishably from one another without much attention or confusion. We briefly describe some of these variations and we also present greedy algorithms that achieve Optimal solutions.

5.5.1 Same Loads

An interesting situation arises from the fact that the loads might be all the same for every advertiser and the capacities may differ. This reflects situations in which there is no reason for uneven client preference distribution, but the capacity of each advertiser is different due to different tolerance for example to overtime schedule and extra working hours. We will show here that a greedy algorithm finds the optimal solution. The ER-k-Same Loads Problem is defined formally by the following :

INPUT: A non-negative rational number L for the common load and a list of n non-negative rational numbers C_i indicating the capacity of each advertiser. We suppose $C_i > L, \forall i$, that is no advertiser is failing initially at its own load. We are also given an integer k which represents the number of attacks.

OUTPUT: The maximum damage we can cause to the system with the k available attacks, that is which nodes to attack to destroy the maximum number of nodes in total. (a similar question would be to find out which is the minimum number k of attacks in order to destroy the system)

In the above scenario the algorithm that we call $max - C$-Greedy finds the optimal solution. We give the proof to the following theorem:

Theorem 5.5.1. *The max-C-Greedy Algorithm is optimal for the ER-k-Same Loads Problem.*

Proof. First of all, we are in the ER-scheme, so the sequence of attacks doesn't affect the final state of the system; only the set A of attacks is important. (see Observation C in the Attack Projection Algorithms subsection). Consequently, the attacks are supposed to be sorted in the capacity sizes. The *max-C*-Greedy Algorithm attacks the highest capacity first (in case of tie we resolve them arbitrarily). Since all the loads are the same, our attack will not define, at least for the first step, the amount of load that will be redistributed (it is always $L/(n-1)$). However, what we can try to achieve is to accelerate the process of cascading failures. The *max-C*-Greedy Algorithm achieves that: the container with C_{max} will also have fs_{max} ($fs_i = C_i - L_i$), so we minimise the total empty space of the system by the most possible at each attack. We can see why the *max-C*-Greedy Algorithm is optimal by an exchange argument: Let's say that the sequence of attacks for an optimal (we denote it by OPT) algorithm is: $OPT_1, OPT_2, ..., OPT_k$ and for our *max-C*-Greedy Algorithm is: $G_1, G_2, ..., G_k$ and that these two algorithms differ at least on one decision they make for the sequence of their k available attacks. Let's denote OPT_j and G_j the first point in the two sequences that a different decision is taken. Up until this point, all the choices are the same and so the state of the system is identical for the two algorithms. Let m the number of remaining containers at this point. We know, by the greedy choice, that $C_{G_j} > C_{OPT_j}$ (if they are equal the two containers are indistinguishable). Immediately after the attack, from the initially remaining m containers, $m-2$ (excluding G_j and OPT_j) of them are exactly the same in both cases. However, in the greedy case we are left with OPT_j remaining and in the OPT case we are left with G_j and so the cascading effect may be more intense for Greedy than the OPT, since the OPT_j has less free space and is more likely to be destroyed more easily. We conclude that by exchanging the OPT_j with G_j the OPT sequence will not deteriorate (meaning destroy fewer containers); the Greedy is at least as good as the OPT and so the greedy finds the optimal solution.

□

5.5.2 Same Free Spaces

Sometimes it might be the case that the containers have no constraints on their load or their capacity but they have a fixed free space. This for example may be the case for power systems that use generators that can only generate a certain amount over their normal usage. We will model the above scenario by the following ER-k-Same Free Spaces Problem and we will give a greedy algorithm that finds the optimal solution. The ER-k-Same Free Spaces Problem is defined formally by the following :

INPUT: A list of n non-negative rational numbers L_i indicating the load of each advertiser and a positive rational number FS indicating the common free space.

OUTPUT: Find out which is the minimum number k of attacks in order to destroy the system.

We changed here the output and instead of having a fixed amount of available attacks trying to inflict maximum damage, we have the goal of destroying the system with the minimum number of attacks. This is because in the case where every container has the same free space, there are no intermediate cascading failures unless we are on the brink of the complete system destruction. In the above scenario the algorithm that we call $max-L$-Greedy finds the optimal solution. We give, in the same spirit as above, the proof to the following theorem:

Theorem 5.5.2. *The max-L-Greedy Algorithm is optimal for the ER-k-Same Loads Problem.*

Proof. First of all, we are in the ER-scheme, so the sequence of attacks doesn't affect the final state of the system; only the set A of attacks is important. (see Observation C in the Attack Projection Algorithms subsection). Consequently, the attacks are supposed to be sorted in the load sizes. The max-L-Greedy Algorithm attacks the highest load first (in case of tie we resolve them arbitrarily). We must observe that since all the free spaces are the same the cascading failures will only occur once and that is only when the system is going to be totally destroyed. So we just need to create a minimum cardinality attack set A with total load greater than the remaining free spaces. The max-L-Greedy Algorithm achieves just that; we give an exchange argument similar to the one used above: Let's say that the sequence of attacks for an optimal (we denote it by OPT) algorithm is: $OPT_1, OPT_2, ..., OPT_k$ and for our max-L-Greedy Algorithm is: $G_1, G_2, ..., G_l$ $(l \geq k)$ and that these two algorithms differ at least on one decision they make for the sequence of their attacks. Let's denote OPT_j and G_j the first point in the two sequences that a different decision is taken. Up until this point, all the choices are the same and so the state of the system is identical for the two algorithms. Let m the number of remaining containers at this point. We know, by the greedy choice, that $L_{G_j} > L_{OPT_j}$ (if they are equal the two containers are indistinguishable). Immediately after the attack, from the initially remaining m containers, $m-2$ (excluding G_j and OPT_j) of them are exactly the same in both cases. However, in the greedy case we are left with OPT_j remaining and in the OPT case we are left with G_j and so the cascading effect may be more intense for Greedy than the OPT, since the G_j has more load. We conclude that by exchanging the OPT_j with G_j the length of the OPT sequence will either not change or shorten; the Greedy is at least as good as the OPT and so the greedy finds the optimal solution. $\square$

5.5.3 Capacities Proportional to Loads

The capacities and the loads in many situations are not fixed, but are somehow related and it is sometimes rational to think that the capacity of a node is proportional to its load rather than completely arbitrary. This is actually true in many power systems, where each line that has a load L, has a capacity not arbitrary apart from L, but rather $C = (1 + \alpha)L$, where α is the tolerance parameter as we have seen in previous chapters. In a social market, the α constant may depict the advertiser's tolerance for overtime which may be proportional to his/her load of work. For example, if we think of a store that is expecting an amount of clients L, then it might as well be prepared for an amount of clients $C = (1 + \alpha)L$. In this variation, we will also show that there is a greedy algorithm achieving the optimal solution. The ER-k-($C \propto L$) Problem is defined formally by the following :

INPUT: A list of n non-negative rational numbers L_i indicating the load of each advertiser and a positive (so that no advertiser is failing initially at its own load) rational number α as the tolerance parameter.

OUTPUT: The maximum damage we can cause to the system with the k available attacks, that is which nodes to attack to destroy the maximum number of nodes in total.
In the above scenario the algorithm that we call $max - L$-Greedy finds the optimal solution. We give, in the same spirit as above, the proof to the following theorem:

Theorem 5.5.3. *The max-L-Greedy Algorithm is optimal for the ER-k-($C \propto L$) Problem.*

Proof. First of all, we are in the ER-scheme, so the sequence of attacks doesn't affect the final state of the system; only the set A of attacks is important. (see Observation C in the Attack Projection Algorithms subsection). Consequently, the attacks are supposed to be sorted in the load sizes. The *max*-L-Greedy Algorithm attacks the highest load first (in case of tie we resolve them arbitrarily). We must observe that since the capacities are proportional to the loads then the container with the maximum load will also have the maximum capacity and also the maximum free space, $fs = C_i - L_i = (1 + \alpha)L_i - L_i = \alpha L_i$. So it seems that the *max*-L-Greedy Algorithm is the best choice: we give an exchange argument similar to the one used above: Let's say that the sequence of attacks for an optimal (we denote it by OPT) algorithm is: $OPT_1, OPT_2, ..., OPT_k$ and for our *max*-L-Greedy Algorithm is: $G_1, G_2, ..., G_k$ and that these two algorithms differ at least on one decision they make for the sequence of their attacks. Let's denote OPT_j and G_j the first point in the two sequences that a different decision is taken. Up until this point, all the choices are the same and so the state of the system is identical for the two algorithms. Let m the number of remaining containers at this point. We know, by the greedy choice, that $L_{G_j} > L_{OPT_j}$ (if they are equal the two containers are indistinguishable) and this means that the damage caused by greedy is at least as severe as OPT's. Immediately after the attack, from the initially remaining m containers, $m - 2$ (excluding G_j and OPT_j) of them are exactly the same in both cases. However, in the greedy case we are left with OPT_j remaining and in the OPT case we are left with G_j

remaining and so the cascading effect may be more intense for Greedy than the OPT, since we destroyed the G_j that had more load and also more free space (more free space means that it will be destroyed later than OPT_j unless attacked - see Observation A in the Attack Projection Algorithms subsection). We conclude that by exchanging the OPT_j with G_j the damage caused to the system may be even greater; the Greedy is at least as good as the OPT and so the greedy finds the optimal solution. $\square$

Observation: As we have seen in a previous chapter (Robustness under Random Failures) it might be useful to consider cases where $C_i = (1 + \alpha_i)L_i$, with L_i, α_i arbitrary. However, this case is identical to the ER-k Problem by choosing the right value for $\alpha_i = C_i/L_i - 1$.

5.6 Hardness Reductions

First we will prove that ER-k-min$\sum L$ Problem is NP-Complete. Then we will prove that the ERAN-k Problem is also NP-complete and that it is also hard to approximate within any non-trivial factor. We will use the k-Subset Sum variant ("Given a set of integers and a target sum S, is there any subset of **size k** whose sum is S?") as defined in the Preliminaries Section.

Theorem 5.6.1. *The ER-k-min$\sum L$ Problem is NP-Complete.*

Proof. Firstly, our ER-k-min$\sum L$ Problem is in NP: The certificate is a list of the k containers we need to attack. We can check in polynomial time (look at ER attack projection algorithm) whether the system is destroyed or not. Since we have a certificate that can be checked in polynomial time, ERP-k is in NP! Given an instance of the k-Subset Sum problem we will create an instance of the decision version of ER-k-min$\sum L$ Problem as we describe next: Let suppose that the set of n integers is $a_1, a_2, ..., a_N$, the numbers we choose are exactly k and that the target integer S for the k-Subset Sum problem. First of all, we can check if the sum of all a's equals to S and respond accordingly. From now on, we suppose $k < n$. We create n containers with loads $L_1 = a_1, L_2 = a_2, ..., L_n = a_n$. We set the load parameter to be the target sum S, that is we set $L = S$. The number of attacks k is the same as the Subset Sum and the parameter k' equals to n, that is we want to destroy all of the containers. The only thing that remains yet undefined is the capacity of each container. We set $C_i = L_i + fs$ where the free space is $fs = \frac{L}{n-k} = \frac{S}{n-k}$. The construction of the instance for the decision ER-k-min$\sum L$ Problem is now over. We chose to enforce all the containers to have the same free space because we prevent the system from having cascading failures unless it is the case that it will be completely destroyed. This is due to the equal redistribution assumption; either all containers will be destroyed by the redistribution or none of them. We observe that in order to have a k size attack set A to destroy the system we must have that $\sum_A L_i \geq (n-k) \cdot fs$ because, by equal redistribution, the remaining containers/advertisers will fail. But the quantity $(n-k) \cdot fs = (n-k) \cdot \frac{S}{n-k} = S$. So we conclude that our solution to ER-k-min$\sum L$ Problem will be a set A with $\sum_A L_i \geq S$ and simultaneously $\sum_A L_i \leq S$, that is $\sum_A L_i = S$ and thus we would have solved the k-Subset

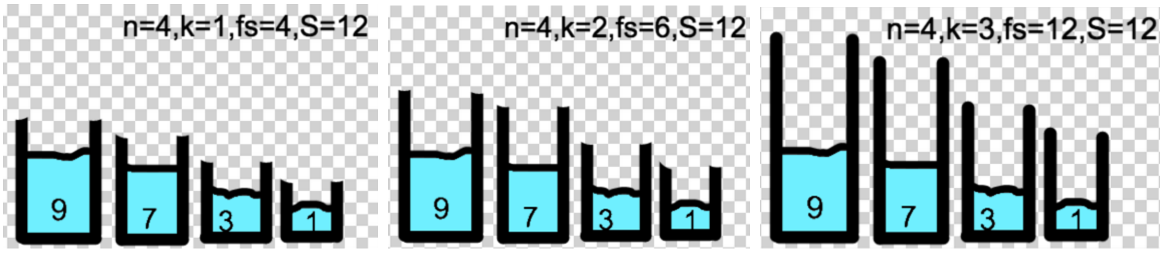


Figure 5.9: *An extensive example for the reduction where the target sum $S = 12$ and the numbers are $(1, 3, 7, 9)$. We start by setting $k = 1$. There is no solution. Then, we set $k = 2$. There is the solution by attacking the container with load 9 and with load 3. Remember, we had set $L = 12$ and $fs = \frac{12}{2} = 6$, so we get a solution for the subset sum. The last picture with $k = 3$ yields no solution.Finally, attacking all the containers $k = 4$ is not a solution as well.*

Sum instance.
We observe that if the Subset-Sum instance is a YES instance (let the solution be the numbers $n_1, ..., n_k$ that sum up to S) then the containers that correspond to the numbers $n_1, ..., n_k$ form a solution to the ER-k-min $\sum L$ Problem. For the reverse direction, assume that the ER-k-min $\sum L$ Problem has solution which includes k containers/advertisers $Cont_1, ..., Cont_k$. Then the loads of these containers constitute a solution to the Subset-Sum problem.

The above reduction can be constructed in polynomial time (linear time to be exact), so if there was a polynomial algorithm that could solve the ER-k-min $\sum L$, then the k-Subset Sum would be in P, which is wrong unless P=NP. Thus, we conclude that the ER-k-min $\sum L$ Problem is NP-complete. We also give an example based on the reduction process for $k = 1, 2, 3$ in Figure 4.

□

Theorem 5.6.2. *The ERAN-k Problem is NP-Complete.*

Proof. The reduction is somewhat direct from the Set Cover Problem if we play a bit with the values of the loads and the capacities. Given an (S, U) instance of the Set Cover Problem with sets $S_1, S_2, ..., S_m$ and elements $u_1, ..., u_n$, the idea of the reduction is to create a network, where we will have representatives of the Sets S_i and of the elements u_j. The graph is bipartite and there is a directed edge from a set S_i to all elements it contains; that is the directed edge $(S_i, u_j) \in E$ if and only if $u_j \in S_i$. Now, we must calibrate the loads and the capacities of the nodes, so that if we attack a node representing a set then all its elements are destroyed due to the cascading failures. One way this can be achieved is by assigning each node S_i to a load of ϵ and arbitrary capacity (it won't matter in this special case for the reduction) and assign each node u_j with a load of 0 and a capacity of ϵ/n, to ensure it will always be destroyed if we choose any set that contains it (actually, we are only interested in the free space for the elements nodes). The construction is now complete. It is done in polynomial time.

We observe that if the Set-Cover instance is a YES instance (let the solution be the sets $S_1, ..., S_k$ that cover the universe of n elements) then the containers that correspond to these sets $S_1, ..., S_k$ form a solution to the ERAN-k Problem (with $k' = n + k$) since an attack to them would mean that all the n elements are destroyed, plus the initially attacked k nodes. For the reverse direction, assume that the ERAN-k Problem has solution which includes k containers/advertisers $Cont_1, ..., Cont_k$ and destroys $k' = n + k$ advertisers in total. Then the same containers $Cont_1, ..., Cont_k$ constitute a Set Cover for the universe U.

□

The most interesting fact, probably, is that the ERAN-k Problem cannot be approximated to any non-trivial factor as the following theorem suggests.

Theorem 5.6.3. *It is NP-hard to approximate the ERAN-k Problem to within a factor of $n^{1-\epsilon}$, where n is the number of containers/nodes, for any $\epsilon > 0$.*

Proof. Suppose we have an (S, U) instance of the Set Cover problem with sets $S_1, S_2, ..., S_m$ and elements $u_1, ..., u_n$. We create the network as shown in Figure 9. Let $\epsilon > 0$. Without loss of generality we can suppose $m < n$. Next, for an arbitrarily large constant c, we add $N = n^c$ more nodes $1, ..., N$ (that's the Group D); each of these nodes is connected to **BB** node and these nodes will either all fail or none depending on the failure of **BB** which only fails if all the u_i's are failed. If there are k sets in Group A that cover all elements, then attacking the nodes corresponding to these k sets will lead to the failure of all the nodes u_i, and thus also all of the Group D . In total, at least $N+n+k$ nodes will be failed. Conversely, if there is no set cover of size k, then no targeted set will succeed in failing all of the u_i, and hence none of the nodes in Group D will fail (unless targeted). In particular, fewer than $n+k$ nodes are failed in the end. If an algorithm could approximate the problem within $n^{1-\epsilon}$ for any ϵ, it could distinguish between the cases where $N + n + k$ nodes are dead in the end, and where fewer than $n+k$ are. But this would solve the underlying instance of Set Cover, and therefore is impossible assuming $P \neq NP$. More practically, given our construction above, imagine we had an algorithm that could approximate in the desired ratio; that means it would output a set A whose attack projection would satisfy: $proj(A) \geq \frac{1}{(n^c + n + m + 1)^{1-\epsilon}} \cdot proj(OPT)$. Now we will compare this output to the number $n + k$, where n is the number of ground elements in U and k is the magnitude of our attack. Let us investigate the following two cases:

i) If $proj(A) \leq n + k$, then we have $n + k \geq proj(A) \geq \frac{1}{(n^c + n + m + 1)^{1-\epsilon}} \cdot proj(OPT)$ which means $proj(OPT) \leq (n+k)(n^c+n+m+1)^{1-\epsilon} < n^c+n+k+1$, for sufficiently large c. From the last inequality we can decide that there is no set cover, since if there was the $proj(OPT) \geq n^c + n + k + 1$.

ii) If $proj(A) > n+k$ then it should be the case that it is a lot greater, given that we chose c sufficiently large, and we decide that there is a set cover, because if there wasn't a set cover then the $proj(A) < proj(OPT) < n+k$.

Because of the gap between $n+k$ and $n^c + n + k + 1$ we could solve the decision Set Cover. Note that our inapproximability result holds in a very simple case of the ERAN-k problem in which not all the parameters L, C of every node had an important role (many loads could be arbitrary, for example, or some of the capacities could be arbitrary). $\square$

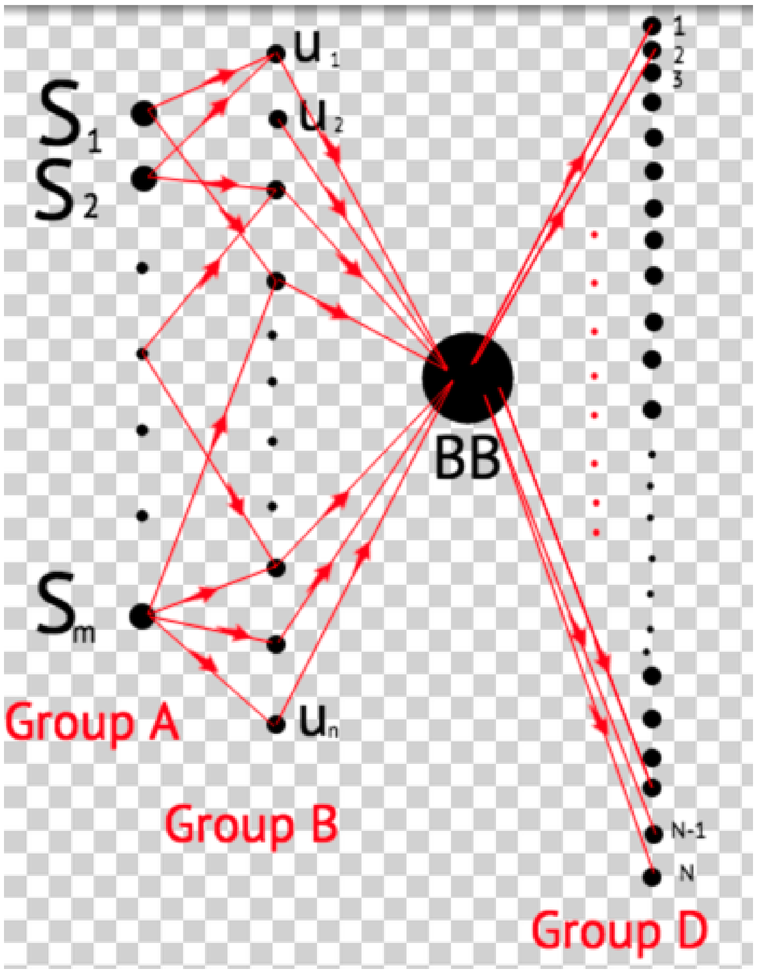


Figure 5.10: *Here is the reduction from the Set Cover in order to prove the hardness of approximation. The loads and the capacities are as follows: For the Group A we have* $(\epsilon, 2\epsilon)$*, for the Group B we have* $(L, L+\frac{\epsilon}{n})$*, for a large constant L, for the* ***BB*** *we have* $(0, nL)$ *and finally for group D we have* $(0, \frac{nL}{n^c})$*. The idea is that the cascading failures proceed to Group D only if we have covered the universe set with at most k sets.*

5.7 Submodularity & k-Attacks

Imagine the following problem with equal redistribution to all that we call k-Attacks Problem (or ER-k (Equal Redistribution-k)):

INPUT: A list of n pairs of non-negative rational numbers in the form (L_i, C_i) indicating the load and the capacity of each advertiser and an integer $0 < k \leq n$. We suppose $C_i > L_i$ that is no advertiser is failing initially at its own load.

OUTPUT: An attack set A of advertisers, with $|A| = k$, so that at the end of the cascading failures the number of failed nodes is maximized.

So, we want to maximise the attack projection of our attack set A, that is $proj(A)$. In other similar situations when, generally, we need to select k elements from a population in order to achieve a maximum overall impact on the rest of the population, we take advantage of the submodularity of the target function. A well-known example of this kind, is the famous influence maximisation problem, in which a greedy algorithm, that takes advantage of the fact that the influence function is submodular in some models (e.g. Independent Cascade, Linear Threshold), gives an (arbitrarily close) approximation guarantee of $1 - 1/e$. One might think that in the problem defined above something similar could help us. However, the submodularity property does not hold for the $proj(A)$. In fact, a knife-edge property might hold: as one adds nodes to attack set A, one initially gets very little cascading failures, but once exactly the right set has been added, the process suddenly spreads very widely. Such a pathological scenario is not possible with the influence function in the Independent Cascade Model and the Linear Threshold Model. And as it is known from the influence maximisation problem, this is actually crucial, since influence functions that grow in a less pathological way, allow for good approximation algorithms. The key to this is the submodularity property. In Figure 10, we give a counterexample for our k-Attacks problem that shows that the submodularity property does not hold. In addition, we can also observe that the attack projection is not supermodular. In the same figure, we give a counterexample for our k-Attacks problem that shows that the supermodularity property does not hold either:

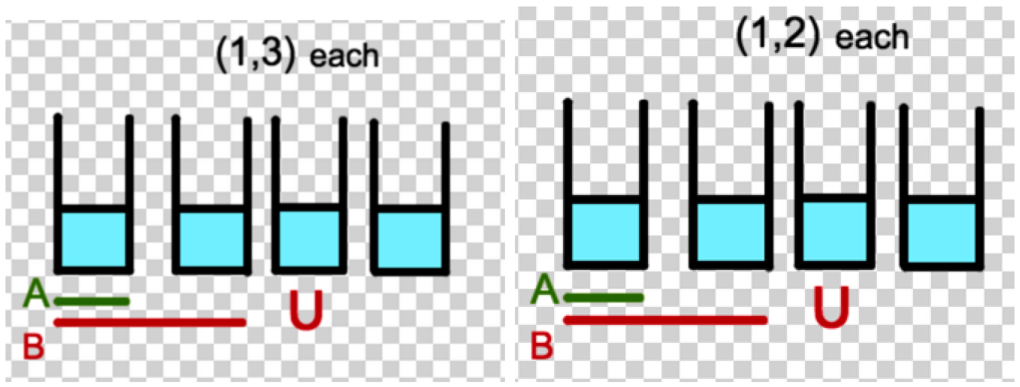


Figure 5.11: *Here are the counterexamples. The first counterexample is for the submodularity: All containers are* $(1,3)$*. The* $proj(A) = 1, proj(B) = 2, proj(A \cup \{v\}) = 2, proj(B \cup \{v\}) = 4$*. Despite the fact that* $A \subset B$ *we get that* $proj(A \cup \{v\}) - proj(A) < proj(B \cup \{v\}) - proj(B)$*, which is the opposite from what the submodularity property would imply. The second counterexample is for the supermodularity: All containers are* $(1,2)$ *in this case. The* $proj(A) = 1, proj(B) = 4, proj(A \cup \{v\}) = 4, proj(B \cup \{v\}) = 4$*. Despite the fact that* $A \subset B$ *we get that* $proj(A \cup \{v\}) - proj(A) > proj(B \cup \{v\}) - proj(B)$*, which is the opposite from what the supermodularity property would imply.*

As a final comment since the k-Attacks problem is a special case of the ERAN-k problem with the topology being the complete graph on the nodes, we get the same conclusions for the attack projection function in the ERAN-k problem.

Observation: So far we assume rational values for loads and capacities, but since we prove hardness results, of course, these results hold in the case we had real values as well.

Chapter 6

Conclusion

6.1 Summary

In this thesis, we studied the problem of System Robustness from various perspectives.

In the beginning we saw two examples from real systems as a motivation for our problem. One example was about the robustness of a power system under equal load redistribution in the case where we initially remove a p-fraction of the system's nodes. We saw that an abrupt transition from the state of normal functionality towards the state of total collapse can occur. In addition, we saw that by selecting wisely some parameters of our problem we could make this transition smoother: a region below the threshold of total collapse can work as a sign of the imminent destruction warning the manager of the system to take action to prevent the failure. As a result of this research, one interesting conclusion is that by having all loads in the system the same we can delay the destruction of our system, meaning that we would have to initially take out the largest possible amount of nodes compared to any other load distribution.

The second example was the blackout that happened in Italy in the year 2003, affecting the lives of millions of people in Italy and in Switzerland as well. A small failure in Italy's power system lead to cascading failures between its power grid and its communication network that led to the failure of one third of its electricity network. We saw why this disaster took place by studying the problem of interdependencies between two networks. We concluded that although a system may be robust to total collapse, some parts of it may be really vulnerable to attacks. Furthermore, an interesting result was that identifying which nodes to take out from an interdependency network in order to cause total failure is NP-complete when the interdependencies can be unidirectional.

After the motivation, we presented our model and our main problems which were the *minCost*-ER-k problem and the *ERAN*-k problem. We gave examples of how an attacker might choose which nodes to attack, we proved two hardness results for our problems, we gave greedy algorithms that fail in achieving a good result and greedy algorithms in special cases that are optimal.

6.2 Future Directions

There are many related problems we would like to address.

Tweaking the redistribution scheme: We can choose to assign the load of the failed nodes differently than equally among neighbours or among all. We can have a weighted manner of redistributing the loads by giving a failed node's load to any of its neighbours according to some preference rules for the neighbours which can be static or dynamically changing based on the system's behaviour at the time of the failure.

Tweaking the load-capacities relations: We can have different relations for the loads and capacities. Here we studied some cases where they were unrelated numbers or related with the equation $C = (1 + \alpha)L$. The tolerance parameter α may be chosen to be different for each node and drawn from a probability distribution. However, different kind of relations can hold generally or sometimes a probabilistic relation may hold. This happens when for example we may have two different types of cables for the current supply and each cable may have different capacity-load relation meaning that the C_i may be drawn from a probability distribution with minimum value the maximum value of the probability distribution used for the loads.

Values of Loads and Capacities: The values may be completely unrelated given of course that the capacity is always greater than the node's load or they can be drawn from probability distributions rather than being just given integers from the input.

Permanent-Transient failures: Depending on the type of system we have to deal with the failure of a node might be permanent or transient/temporary. This alter the network as nodes fail and then are revived again and interesting questions for the system's evolution arise.

Defender's Perspective: We studied the attacker's perspective, proving that finding a budgeted solution is NP-complete but what about the defender's perspective? The defender might choose the redistribution dynamically if he can take advantage of the node's states, he might be interested in reviving some nodes according to some associated cost and some total budget he may have, he may pay for keeping secret some of the loads or capacities trying to trick the attacker. Further investigation is needed to understand how the system may evolve in such scenarios. Game theoretic aspects are needed to be studied as well if for example the attacker and the defender make a move in turns having particular goals as for the final state of the system.

Bibliography

[1] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S.Porcellinis, R. SEtola *Modelling interdependent infrastructures using interacting dynamical models* , 2008.

[2] David Kempe, Jon Kleinberg, Eva Tardos, *Maximizing the Spread of Influence through a Social Network* , 2003.

[3] V. Vazirani, *Approximation Algorithms*

[4] S. Dhakal, M. Hayat, J. Pezoa, C. Abdallah, J. Birdwell, J. Chiasson, *Load Balancing in the Presence of Random Node Failure and Recovery*

[5] Jon Kleinberg, *The Convergence of Social and Technological Networks*

[6] D. Nguyen, Y. Shen, M. Thai, *Detecting Critical Nodes in Interdependent Power Networks for Vulnerability Assessment*

[7] A. Arulselvan, C. Commander, O. Shylo, P. Pardalos, *Cardinality-Constrained Critical Node Deletion Problem*

[8] Alexander Veremyev, Vladimir Boginski, Eduardo Pasiliao, *Exact Identification of critical nodes in sparse networks via new compact formulations*

[9] Jon Kleinberg, *The Small-World Phenomenon: An Algorithmic Perspective*

[10] J. Aspnes, Kevin Chang, A. Yampolskiy *Inoulation Strategies for Victims of Viruses and the Sum-of-Squares Partition Problem*

[11] G. Tuli, C. J. Kuhlman, S. S. Ravi, D. J. Rosenkrantz, *Blocking Complex Contagions Using Community Structure*

[12] D. Zhou, H. Stanley, G. D' Agostino, A. Scala, *Assortativity decreases the robustness of interdependent networks*

[13] X. Peng, H. Yao, J. Du, Z. Wang, C. Ding, *Load-induced cascading failures in interconnected networks*

[14] M. E. J. Newman, *The spread of epidemic disease on networks*

[15] M. E. J. Newman, S. H. Strogratz, D. J. Watts, *Random Graphs with arbitrary degree distributions and their applications*

[16] Jon Kleinberg, *Cascading Behavior in Networks: Algorithmic and Economic Issues*

[17] Marzieh Parandehgeibi, Eytan Modiano *Robustness of Interdependent Networks: The case of communication networks and the power grid* 2013.

[18] A. Veremyev, A. Sorokin, V. Boginski, E. Pailiao, *Minimum Vertex Cover problem for coupled interdependent networks with cascading failures*

[19] Mario Ventresca, Dionne Aleman, *Efficiently Identifying critical nodes in large complex networks*

[20] S. Buldyrev, R. Parshani, Gerald Paul, H. Eugene Stanley, S. Havlin, *Catastrophic cascade of failures in interdependent networks*

[21] S. Buldyrev, Nathaniel W. Shere, G. Cwilich, *Interdependent Networks with identical degrees of mutually dependent nodes*

[22] C. Kuhlman, G. Tuli, S. S. Ravi, *Blocking Simple and Complex Contagion by edge removal*

[23] Jianxi Gao, Shlomo Havlin, *Robustness of network of networks under targeted attack*

[24] D. Callaway, M. E. J. Newman, S. Strogratz, D. Watts, *Network Robustness and fragility: Percolation on random graphs*

[25] Shlomo Havlin, *The extreme vulnerability of interdependent spatially embedded networks*

[26] Silvano Martello, Paolo Toth, *Knapsack Problems*

[27] Xinran He, David Kempe, *Stability of Influence Maximization*

[28] Yang Wang, D. Chakrabarti, Chenxi Wang, Christos Falloutsos *Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint*

[29] O. Yagan, D. Qian, J. Zhang, D. Cochran, *Information Diffusion in Overlaying Social-Physical Networks*

[30] O. Yagan, D. Qian, J. Zhang, D. Cochran, *Conjoining Speeds up Information Diffusion in Overlaying Social-Physical Networks*

[31] George Giakkoupis, A. Gionis, E. Terzi, P. Tsaparas *Models and Algorithms for Network Immunization*

[32] David Kempe, Jon Kleinberg, Eva Tardos, *Influential Nodes in a Diffusion Model for Social networks*

[33] Hyang-Won Lee, Eytan Modiano, *Diverse Routing in Networks with Probabilistic Failures*

[34] Vartika Bhandari, Nitin H. Vaidya *Reliable Broadcast in Wireless Networks with Probabilistic Failures*

[35] Osman Yagan, *Robustness of power systems under a democratic fiber bundle-like model*, 2015.

[36] O. Yagan, D. Qian, J. Zhang, D. Cochran, *Optimal Allocation of Interconnecting Links in Cyber-Physical Systems: Interdependence, Cascading Failures and Robustness*

[37] Aris Pagourtzis, Paolo Penna, K. Schlude, K. Steinhofel, D. S. Taylor, P. Wildmayer *Server Placements, Roman Domination and Other Dominating Set Variants*

[38] J. Aspnes, Kevin Chang, A. Yampolskiy *Inoulation Strategies for Victims of Viruses and the Sum-of-Squares Partition Problem*

[39] G. Tuli, C. J. Kuhlman, S. S. Ravi, D. J. Rosenkrantz, *Blocking Complex Contagions Using Community Structure*