Vulnerability and Protection Analysis of Critical Infrastructure Systems

by

Joydeep Banerjee

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2017 by the
Graduate Supervisory Committee:

Arunabha Sen, Chair
Partha Dasgupta
Guoliang Xue
Gurulingesh Raravi

ARIZONA STATE UNIVERSITY

August 2017

ABSTRACT

The power and communication networks are highly interdependent and form a part of the critical infrastructure of a country. Similarly, dependencies exist within the networks itself. Owing to cascading failures, interdependent and intradependent networks are extremely susceptible to widespread vulnerabilities. In recent times the research community has shown significant interest in modeling to capture these dependencies. However, many of them are simplistic in nature which limits their applicability to real world systems. This dissertation presents a Boolean logic based model termed as Implicative Interdependency Model (IIM) to capture the complex dependencies and cascading failures resulting from an initial failure of one or more entities of either network.

Utilizing the IIM, four pertinent problems encompassing vulnerability and protection of critical infrastructures are formulated and solved. For protection analysis, the Entity Hardening Problem, Targeted Entity Hardening Problem and Auxiliary Entity Allocation Problem are formulated. Qualitatively, under a resource budget, the problems maximize the number of entities protected from failure from an initial failure of a set of entities. Additionally, the model is also used to come up with a metric to analyze the Robustness of critical infrastructure systems. The computational complexity of all these problems is NP-complete. Accordingly, Integer Linear Program solutions (to obtain the optimal solution) and polynomial time sub-optimal Heuristic solutions are proposed for these problems. To analyze the efficacy of the Heuristic solution, comparative studies are performed on real-world and test system data.

Owing to some limitations of the IIM, the dissertation also introduces an extended version of the model termed as Multi-scale Implicative Interdependency Relation (MIIR) model. Utilizing the MIIR model, the K Contingency List problem is solved with respect to the power network. The problem solves for a set of K entities which when failed would maximize the number of previously healthy entities to fail eventually. Owing to the problem being NP-complete, a Mixed Integer Program (MIP) to obtain the optimal solution and a polynomial time sub-optimal heuristic are provided. The efficacy of the heuristic with respect to the MIP is compared by using different test system data.

To Ma and Baba

ACKNOWLEDGMENTS

Firstly and most importantly, I would like to thank my academic adviser Dr. Arunabha Sen without whom this dissertation would not have been possible. He introduced me to the beautiful world of combinatorial algorithms and optimization. He always encouraged and motivated me to work on different challenging problems. It was an immense pleasure for me to work under his guidance.

Secondly, I would like to thank my dissertation committee members from ASU, Dr. Guoliang Xue and Dr. Partha Dasgupta. They were extremely accommodating to my requests and provided valuable feedbacks on my research. They also incredibly honed my intellectual thinking ability in the courses they taught. I had a great experience and pleasure to intern under Dr. Gurulingesh Raravi, my committee member from Xerox Research. His out of the box ideas and constant motivation has been a source of encouragement and positive thinking for me.

I would like to thank my lab members Chenyang Zhou, Kaustav Basu and Arun Das with whom I worked on different interesting problems. I would also like to thank Dr. Anamitra Pal with whom I worked on the last problem of my dissertation.

I would also like to thank my friends — Subhadip Senapati, Kaushik Sarkar, Anirban Samanta, Suratna Hazra, Tathagata Chakraborti, Sayantan Das, Shatabdi Roy Chowdhury, Pritha Bisharad and Soumajyoti Sarkar for making my stay at Tempe and Ph.D. life memorable. The same goes for my fellow Jadavpur University friends — Malay Chakrabarti, Sk Minhazul Islam, Saurav Ghosh and Pradipta Ghosh.

Lastly, I am forever indebted to my mother, father, and grandmother for supporting and encouraging me as always. I am also extremely thankful to my girlfriend Akanksha for being a very important person in my life. She was always there to guide me in handling different topsy-turvy scenarios and standing by my side.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Critical Infrastructures of a nation like Power, Communication, Transportation Networks etc. exhibit strong intra-network and inter-network dependencies to drive their functionalities. The symbiotic relationship that exists between Power and Communication network provides an example of the inter-network dependency. To elaborate this further, consider entities of either network: electricity generation and power flows are partially controlled by the Supervisory Control and Data Acquisition (SCADA) system through signals received from Remote Terminal Units (RTUs). Meanwhile, every communication network entity involved in sensing, sending and controlling the power grid are dependent on power network entities to ensure their successful operation. Owing to these dependencies, failure of some entities in either network may eventually result in a cascading failure that can cause a widespread blackout in the combined system.

Additionally, intra-network dependencies exist as well in a critical infrastructure and is described with the help of power network. In an abstract level, a power network is composed of the following entities — Generation Bus, Load Bus, Neutral Bus (or zero injection bus) and Transmission Lines. When a transmission line trips, the power flowing through the transmission lines needs to be redirected to satisfy load demand of the load buses. This may cause the power flow in some other transmission line to go beyond its line capacity causing it to trip. Eventually, these failures might result in a cascade of trippings/failures resulting in a blackout. Cascading failures in power and/or communication network due to intra/inter dependencies have disastrous effects as seen in power blackouts which occurred in New York (2003) [1], San Diego (2011)

[2] and India (2012) [3]. Thus modeling these dependencies is critical in understanding and preventing such failures which might be triggered by natural as well as man-made attacks.

In the last few years, there has been considerable activity in the research community to study Critical Infrastructure Interdependency. One of the earliest studies on robustness and resiliency issues related to Critical Infrastructures of the U.S. was conducted by the Presidential Commission on Critical Infrastructures, appointed by President Clinton in 1996 [4]. Rinaldi et al. are among the first group of researchers to study interdependency between Critical Infrastructures and to propose the use of complex adaptive systems as models of critical infrastructure interdependencies [5], [6]. Pederson et al. in [7], provided a survey of Critical Infrastructure Interdependency modeling, undertaken by U.S. and international researchers. Motivated by the power failure event in Italy 2003, Buldyrev et al. in [8], proposed a graph-based interdependency model, where the number of nodes in the power network was assumed to be the same as the number of nodes in the communication network, and in addition there existed a one-to-one dependency between a node in the power network to a node in the communication network. The authors opine in a subsequent paper [9] that the assumption regarding one-to-one dependency relationship is unrealistic and a single node in one network may be dependent on multiple nodes in the other network. Lin et al. presented an event driven co-simulation framework for interconnected power and communication networks in [10], [11]. A game theoretic model for a multi-layer infrastructure networks using flow equilibrium was proposed in [12].

As discussed above, a number of models have been proposed that capture the dependencies in critical infrastructure systems [8], [9], [12], [13], [14], [15], [16], [17]. However, each of these models has their own shortcomings and a survey of these models along with a detailed analysis of their functionality is presented in Chapter 2.

Authors in [18] bring out the need to address the complex dependency which can be explained through the following example. Let a_x (which can be a generator, substation, transmission line etc.) be a power network entity and b_w, b_y, b_z (which can be a router, end system etc.) a set of communication network entities. Consider the dependency where the entity a_x is operational if (i) entities b_w and (*logical AND*) b_y are operational, or (*logical OR*) (ii) entity b_z is operational. Models in [8], [13], [9], [14], [12], [15], [16], [17] fails to capture this kind of dependency. Motivated by these findings and limitations of the existing models, the authors in [18] proposed a *Boolean logic based dependency* model termed as *Implicative Interdependency Model* (IIM). For the example stated above, the dependency of a_x on b_w, b_y, b_z can be represented as $a_x \leftarrow b_w b_y + b_z$. This equation representing the dependency of an entity is termed as *Interdependency Relation* (IDR).

The IIM model forms the basis of this dissertation. Chapter 3 provides a more detailed description along with the test data sets created using this model. The IIM model is generic enough to capture the interdependencies in an interdependent network as well as intra-network dependencies. The test data sets created in Chapter 3 are subdivided into two groups — (a) data sets capturing dependencies in power network, and (b) data sets capturing dependencies in inter-dependent power and communication network. Using these test data sets, the performance of different solutions proposed for the problems in Chapters 4 - 7 of this dissertation are measured.

Chapters 4 - 7 discusses different problems and their solutions related to vulnerability and protection analysis of critical infrastructure systems. These set of problem utilizes the IIM model in their problem formulation and solutions. The first two problems use the concept of *hardening*. An entity when hardened is considered to be exempted from any kind of attack and can sustain itself without any dependency on other entities. The Entity Hardening Problem is introduced in Chapter 4. It describes a situation where an operator, with a limited budget, must decide which entities to harden, which in turn would minimize the damage, provided a set of entities fail initially. The Targeted Entity Hardening problem is discussed in Chapter 5 which is a restricted version of the Entity Hardening problem. This problem presents a scenario where the protection of certain entities is of higher priority. If these entities were to be nonfunctional, the economic and societal damage would be higher when compared to other entities being nonfunctional.

Modifying dependencies by adding additional dependency implications using entities (termed as *auxiliary entities*) is shown to mitigate the issue of vulnerability in intra/inter dependent systems to a certain extent. With this finding, the Auxiliary Entity Allocation problem in introduced in Chapter 6. The objective of this problem is to maximize protection in Power and Communication infrastructures using a budget in the number of dependency modifications using the auxiliary entities. Chapter 7, a new metric of robustness using the IIM model is defined for inter/intra dependent critical infrastructure systems. All the four problems are proved to be NP-complete. For each of these problems, Integer Linear program to obtain the optimal solution is proposed along with a sub-optimal heuristic with polynomial time complexity. The test data sets are used to measure the efficacy of the heuristic solutions compared to the Integer Linear program.

In the course of research, the IIM model is seen to have its own limitations. The IDRs (i.e. dependency equations) form the core of the IIM model. Even though a primary approach to create these dependency equations is shown in Chapter 3, they are based on certain assumptions which might limit their applicability to real world problems. Finding a generic technique to have a near accurate abstraction of an intra/inter dependent critical infrastructure system is seen to be difficult using the IIM model. To address this limitation, in Chapter 8 the Multi-scale Implicative Interdependency Relation (MIIR) model is proposed primarily focusing on intra-dependent power network. Along with the IDRs, the MIIR model also takes into consideration the power values (i.e. generation of a generator bus, power flowing through transmission lines etc.) of different entities in the system. These power values can be obtained from Phasor Measurement Units (PMU) associated with the entities (there has been considerable research on using PMU data in mitigating failure in power networks [19], [20], [21], [22]). A formal description of the model along with its working dynamics and a brief validation with respect to the 2011 Southwest Blackout are provided. Utilizing the MIIR model, the K Contingency List problem is proposed. For a given time instant, the problem solves for a set of K entities in a power network which when failed at that time instant would cause the maximum number of previously healthy entities to fail eventually. Owing to the problem being NP-complete a Mixed Integer Program (MIP) is devised to obtain the optimal solution and a polynomial time sub-optimal heuristic. The efficacy of the heuristic with respect to the MIP is compared by using different power network test system data.

The dissertation is concluded in Chapter 9 with discussions on possible research problems that can be pursued based on this research.

Chapter 2

A SURVEY OF INTER-DEPENDENCY MODELS IN CRITICAL INFRASTRUCTURE SYSTEMS

In the last few years there has been an increasing awareness in the research community that the critical infrastructures of the nation do not operate in isolation. In fact, they are closely coupled with other infrastructures such that the well being of one infrastructure depends heavily on the well being of another. As an example, consider the interdependent relationship between the power, communication, and transport networks as shown in Figure 2.1 ([6]).

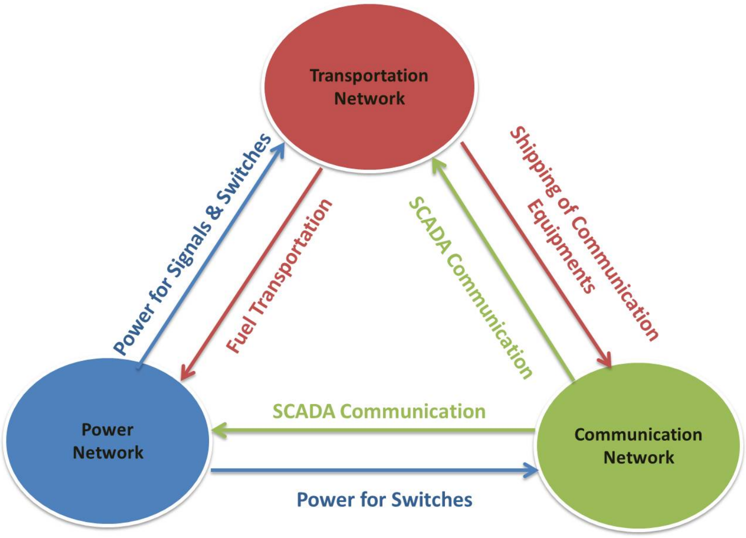

Figure 2.1: Power, Communication and Transportation Network Interdependency

If we focus exclusively on the power and communication networks we observe that entities of the power grid, such as the Supervisory Control and Data Acquisition (SCADA) systems, that control power stations and sub-stations, are dependent on the communication network to receive their operational commands. While entities of the communication network, such as routers and cell towers, are dependent on the power grid to remain operational. Compounding the complexity of analysis of this symbiotic relationship between the two networks, is the effect of cascading failures across these networks. For instance, not only can entities of the power networks, such as generators and transmission lines, trigger a power failure, but also communication network entities, such as routers and optical fiber lines, can trigger failures in the power grid. Thus, it is essential that the interdependency between different types of networks be understood well, so that preventive measures can be taken to avoid cascading catastrophic failures in such multi-layered network environments.

With the continued focus for developing realistic failure propagation models that aid in analyzing, and mitigating the effects of cascading faults across the entities of the multi-layered network, several failure propagation models have been studied that address the interdependency relationship between power, and communication networks ([8, 14, 16, 15]), and space based networks ([23]).

A brief survey of the existing interdependency models for critical infrastructure networks that have been proposed in the is presented. The chapter then address the considerations that need to be taken into account for capturing the complex interdependency that exists between power grid and communication networks in the real world.

2.1 Interdependency Models

2.1.1 Buldyrev et al. Interdependency Model

Motivated by the electricity blackout in Italy (2003) ([24]) [8] proposed a cascading failure model for interdependent networks. The power and communication infrastructures can be represented as networks. These networks are depicted as two connected graphs P (for power network) and C (for communication network) with same number of nodes. To represent the interdependency between the networks, bidirectional links between P and C, termed as $P \leftrightarrow C$ edges, are considered with every node in each graph connected to exactly one node in the other graph as shown in Figure 2.2(a). In Figure 2.2(a) the interdependent network shown consists of power network nodes p_1, p_2, p_3 and p_4 and communication network nodes c_1, c_2, c_3 and c_4. Blue and green edges denote intra links in power and communication network respectively and black edges denote the interdependency (inter links). These inter links represent the interdependency relationship that a node in the power network is dependent on exactly one node in the communication network and vice-versa. Thus capturing the fact that a failure of a node in the power (communication) network causes the corresponding node in the communication (power) network to fail.

Failures are considered in the model when a fraction of the nodes from any of the two graphs P, or C are removed. Upon the introduction of a failure in the graph P, the failed nodes are removed and correspondingly, the nodes in the graph C that are connected via $P \leftrightarrow C$ edges to the attacked nodes are also removed. Parallel to the node removals, any edge within graph P or C, or $P \leftrightarrow C$ edges that do not have one node at each end point are also simultaneously removed.

The cascade now proceeds as follows. In the first stage, the set of connected components in the graph P is defined as P_1 clusters. The set of C nodes connected to the P_1 clusters by $P \leftrightarrow C$ edges are termed as C_1 sets. Any edges in graph C, that connects these C_1 sets are removed. The set of connected components in graph C after this removal of edges are defined as C_2 clusters. In the second stage using same procedure as that to find the c_2 cluster and C_1 sets, P_2 sets from C_2 clusters and P_3 clusters are obtained. In subsequent stages this cascade process then oscillates between the two graphs until a *steady state* is reached when no further removal of edges in the graphs are possible. This is described through an example in Figure 2.2(a-c). The node p_1 is attacked and removed from the graph P thus resulting in P_1 clusters consisting of two connected components having nodes $\{p_1\}$ and $\{p_2, p_4\}$ respectively. Node c_3 in graph C fails due to its interdependency with node p_3. The corresponding C_1 sets obtained due to these failures are $\{c_1\}$ and $\{c_2, c_4\}$. The edge (c_1, c_4) is removed as it connects the two sets in C_1. The corresponding C_2 clusters consist of connected components $\{c_1\}$ and $\{c_2, c_4\}$. The failure then reaches a steady state as no edges can be removed in the next stage.

At the steady state, the interdependent network consists of *mutually connected clusters.* Each mutually connected cluster consists of nodes having the properties (a) the nodes in graphs P and C are completely connected, (b) each of these nodes which belong to the graph P (C) has $P \leftrightarrow C$ edge with graph C (P). Note that there exists no intra-links between any of the mutually connected clusters. In Figure 2.2(c) the mutually connected clusters are thus $\{p_1, c_1\}$ and $\{p_2, p_4, c_2, c_4\}$.

The *largest mutually connected cluster* is defined as the one having the maximum number of nodes (cluster $\{p_2, p_4, c_2, c_4\}$ in the example). Given a fraction $1 - p$, $(0 \leq p \leq 1)$ of nodes that are removed from the interdependent network (due to a failure), the ratio P_∞ defines the number of nodes in the largest mutually connected cluster at the steady state, as compared to the initial number of nodes in the network. For the purpose of simulation and study, the power and communication networks are considered as, coupled scale free, Erdos Reyni [25], and random networks. Different values of P_∞ were computed by varying the values of p, and the size of the network. It was observed that, above a percolation threshold p_c, the value of P_∞ changes from the neighborhood of zero to the neighborhood of one for a given network size. From this observation the authors infer that when the fraction of failed node is below $1 - p_c$ of the original number of nodes, the largest connected cluster has a size approximately

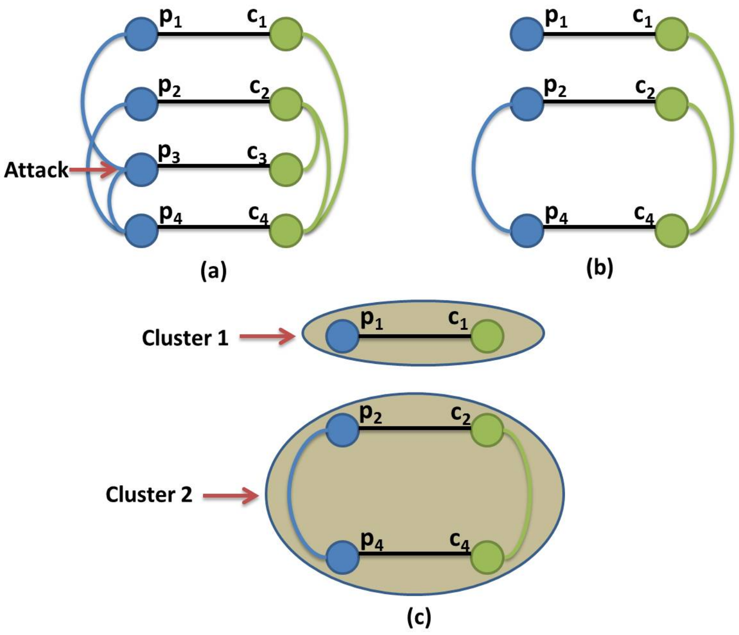

Figure 2.2: Representation of Interdependency and Cascading Failure in Power and Communication network as Demonstrated by Buldyrev et al.

equal to the size of initial pre-failure network. The percolation threshold p_c for Erdos Reyni networks is validated by analytical results.

In subsequent papers, Buldyrev et al. extend their work from their original cascading failure model (as discussed above), to interdependent networks with directional dependency ([9]), and interdependency between more than one network ([13]).

One noticeable shortcoming of this model proposed by Buldyrev et al. is that it does not distinguish between nodes in either network as separate entities. Nodes in the power network may be functionally separate entities such as power plants, sub-stations, and load nodes. Similarly, nodes in the communication network may be functionally separate entities such as cell towers, and routers. When separate entities of the network are considered, the proposed cascading model may not work in the same way as assumed by the authors, and also the dependency relationship of one type of entity to the other may not be able to be captured with this model. Another potential drawback to this model is for the functionality of the mutually connected cluster. The mutually connected clusters generated after the cascade may not be completely functional because of the physical limitations of the network ([17]). For example, the nodes from the power grid in a mutually connected cluster may not be able to provide sufficient power to the nodes in the communication network due to the limits on the power generation capacities. Thus, it would be wrong to assume that the residual mutually connected clusters continue to be functional after a cascade simply because they remain connected.

2.1.2 Rosato et al. Coupling Model

[14] model the power flow in the power grid, and the data flow in the communication network separately. They then analyze the effect of failures in the communication network, caused by failures in the power grid using a coupling model between the two infrastructures. Their analysis of the failure propagation is performed on the backdrop of the Italian high voltage electric transmission network (HVIET), and the high-bandwidth backbone of the Italian Internet network (GARR). Data for both the networks were gathered from documentation available in the public domain.

For modeling the power network, the HVIET network is represented by an undirected graph consisting of three type of vertices, namely, source nodes (nodes that supply power to the network), load nodes (nodes that draw power out of the network), and junction nodes (which neither draw nor supply power to network, but act as relays). The edges of the graph corresponds to the transmission lines. The power flow dynamics in the power grid relies on the DC power flow model as given by [26]. At every occurrence of a failure of one or more nodes, or transmission lines (edges), the power flow dynamics are recalculated using this model. It is to be noted here that the DC power flow model considers the physical constraints pertaining to the maximum power flow possible over a transmission line while computing the minimum load re-dispatch (reducing the power drawn out by the load nodes) after a failure. The authors define the quality of service (QoS) of the power network as the ratio of the change in the total power drawn by the load nodes after the failure event, as compared to the total power drawn by the load nodes before the failure event.

For modeling the communication network, the GARR network is represented as a graph consisting of high-bandwidth backbone links as edges, and the Italian universities and research institutions as nodes. For computing the total amount of traffic inflow into the network, the probability that a node generates a packet λ, $(0 \leq \lambda \leq 1)$ is considered at each time step. For each generated packet a random node is chosen as its destination. A probabilistic packet routing model is considered along the lines of [27] for sending the packets to their intended destinations. The average delivery time is defined as the average of the packet transmission time from source to destination over all packets delivered correctly within a particular time interval. The average delivery time is then used as a metric to define the efficiency of the network for a given value of λ.

The coupling between the two networks is achieved by associating a node from the communication network to the closest load node from the power network (Euclidean distance). Note that this coupling is one directional, that is, for a node to be operational in the communication network it is dependent on a node from the power network, but not vice-versa. In a failure event, if a load node i that was initially extracting power P_i^0 units, now extracts P_i units of power after the subsequent load re-dispatching process. The communication nodes coupled to i remain operational as long as the value of P_i is greater than or equal to αP_i^0, $(0 \leq \alpha \leq 1)$. The coefficient α is termed as the strength of coupling between the two networks.

The authors then use the above coupling model to analyze and simulate the effect of random link failures in the power network for a fixed parameter of α (taken as $\alpha = 0.75$). The main insight of their simulation is that even with small failure events in the power (HVIET) network (small with respect to number of transmission lines failed), the communication (GARR) network can get completely disconnected.

The individualized modeling of the power and communication network done by Rostato et al. is realistic to a point, but the coupling model reflects only a one way dependency model and fails to represent the interdependency that exists between power and communication networks of today. This shortcoming may prohibit the accurate cascading failure scenarios when the faults originate from the communication network and cascade through to the power network.

2.1.3 Nguyen et al. Interdependency Model

In [16] propose a cascading model in similar lines of [8], and address the problem for identifying the critical nodes in an interdependent network. In their model, the power network, and communication network are considered as graphs $G_s = (V_s, E_s)$ and $G_c = (V_c, E_c)$, and the interdependency is represented by an unidirectional edge set E_{sc} that connect vertices from set V_s with set V_c in a composite graph containing this edge set, and both the power, and communication networks graphs. A failure due to a dependency relation is outlined by the assumption that, not only do the failed node(s) cease to operate, but also the nodes connected to the failed nodes via edges from the edge set E_{sc} also become non-operational. Failures propagate in the following way: the failed nodes and the incident edges to these nodes that belong to G_s (power network), and G_c (communication network) are removed to generate G'_s and G'_c respectively. Then, the largest connected components L_s and L_c are computed for the graphs G'_s and G'_c. Any node $n_s \in G'_s$ that does not belong to L_s, and any node $n_c \in G'_c$ that does not belong to L_c are considered non-operational. Failures due to the dependency relations are simultaneously considered, and propagation ensues until a *steady state* is reached when no further nodes in either network can fail. An example showing this failure propagation is shown in Figure 2.3 (a-c) where the power network graph consist of nodes p_1,p_2,p_3,p_4 and communication network graph consist of nodes c_1, c_2, c_3, c_4. Blue and green arcs represent edges in power and communication network graph respectively and black arc represent the edges in E_{sc}. A sample failure propagation is described as follows — (a) The node p_3 is attacked. (b) The edges incident on node p_3 are removed due to its failure along with its interdependent node c_3 in communication network and all its associated intra links. (c) The node p_1 and c_1 fails as it is disconnected from the largest connected component in the power network. The steady state is reached with nodes p_2 and p_4 in power network and nodes c_2 and c_4 in communication network as functional nodes after the failure event.

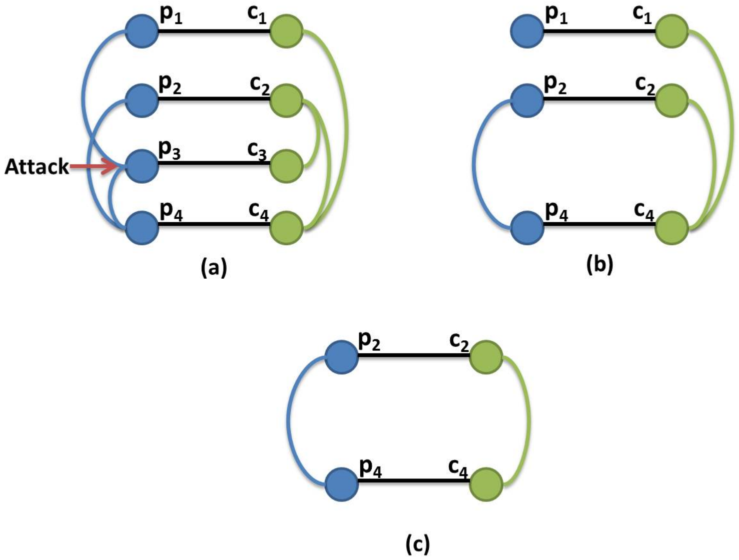

Figure 2.3: Representation of Interdependency and Cascading Failure in Power and Communication Network as Demonstrated by Nguyen et al.

Using the above defined failure propagation model, the authors consider the problem of identifying a set of critical nodes in the power network of size less than a positive integer k, such that at the steady state the size of the largest connected component in the power network is minimized. The authors show that this problem is NP-complete by reduction from the decision version of the Maximum Independent Set problem, and infer that this problem is in-approximable within a bound of $2 - \epsilon$. Three greedy approximation algorithms are proposed by the authors for approximating the solution to this problem in polynomial time, namely, *Maximum Cascade* (Max-Cas), *Iterative Interdependent Centrality* (IIC), and *Hybrid.*

The authors perform an extensive simulation of the proposed algorithms using three different power network, and communication network data sets. The data sets considered were (i) US Western States power network, and a synthetic scale free communication network with an exponential factor, $\beta = 2.2$, (ii) Synthetic scale free power network with $\beta = 3.0$, and a synthetic scale free communication network with $\beta = 2.2$, and (ii) Scale free power and communication networks with the same $\beta = 2.6$. For each of the simulations the interdependency relationship between the two networks were setup using a random weighted permutation of nodes of the two networks.

In the simulations it was observed that the Hybrid algorithm takes lesser time and has better performance bounds than the other two algorithms. In the process of the simulations, it was observed that when interdependent systems are loosely connected they are more vulnerable to failure. Their observations also included that sparse interdependent networks are more vulnerable to cascading failures. This was observed from simulations carried out by varying the exponential factor of the scale free communication network, while keeping the exponential factor of the power network, and the total number of nodes constant. The simulations carried out by the authors by varying the total number of nodes of both the networks, while keeping a fixed exponential factor of the considered scale free networks, showed that large networks are more vulnerable to cascading failures.

The observable shortcomings of this model are similar to the drawbacks discussed above for the model proposed by [8]. Without the distinction of nodes in the networks into separate entities, such as power plants, and substations, for the power network, and cell towers, and routers for the communication network, the failure cascading model may not represent the workings of real world networks. Thus hindering the analysis, and mitigation of faults caused by cascading failures in multi-layer critical infrastructure networks.

2.1.4 Parandehgheibi et al. Interdependency Model

[15] also consider the power and communication infrastructure networks to analyze the effect of cascading failures on these interdependent networks. In their model, the power network graph $P = (V_p, E_p)$ consist of vertices V_p representing the generators, and substations, and edges E_p representing the transmission lines. Similarly, the communication network graph $C = (V_c, E_c)$ consist of vertices V_c representing the control centers, and routers, edges E_c representing the communication lines. In the graphs, it is assumed that nodes represented by generators, and control centers, are *autonomous*, i.e. these nodes operate independently without any dependency on any other node across both the networks. In this model, dependency between network entities is represented by coupling the routers, and substations with edges E (directed or undirected), in a composite graph of $G = (V, E, E_p, E_c), V = V_p \cup V_c$. Whether a node of this composite graph G is operational or not is defined by the following functional rules: If the node represents a substation, it remains operational as long as, (i) there exists a path between the substation and a generator via the power network

edges E_p, and (ii) there exists a path between the substation and a router (to receive control signals) via edges of E. If the node represents a router, it remains operational as long as, (i) there exists a path between the router and a control center via the communication network edges E_c, and (ii) there exists a path between the router and a substation (to receive power) via edges of E. Lastly, if the node represents a generator, or a control center, it remains continuously functional.

At the time of the initial failure (due to a possible attack, or fault), the failed nodes, or edges are removed from the graph G. The failure propagation is then represented in the model by iteratively removing the failed nodes and all their incident edges from graph G that do not satisfy the aforementioned functional rules. This propagation continues until a *steady state* is reached when no further removals of nodes, or edges are necessary. An example of the described failure propagation is illustrated in Figure 2.4. In the figure the power network consists of a generator G and substations s_1, s_2, s_3 and communication network consists of control center C and routers r_1, r_2, r_3. Blue edges denotes the power network edges (composed of transmission lines) and green edges denotes the communication network edges (composed of communication links). Black edges denotes the interdependency between substation of power network and routers in communication network. An example of failure propagation in this model is discussed as follows — (a) The substation s_1 is attacked. (b) Failure of substation s_1 results in removal of all power network edges incident on s_1 and failure of interdependent router r_1 and removal of communication network edges incident on it. (c) Substations s_2, s_3 and routers r_2, r_3 fails and hence are removed as they do not satisfy both the properties for being being functional as mentioned. The edges incident on these substations and routers are subsequently removed. The resultant interdependent network after the failure consists of two autonomous nodes G and C.

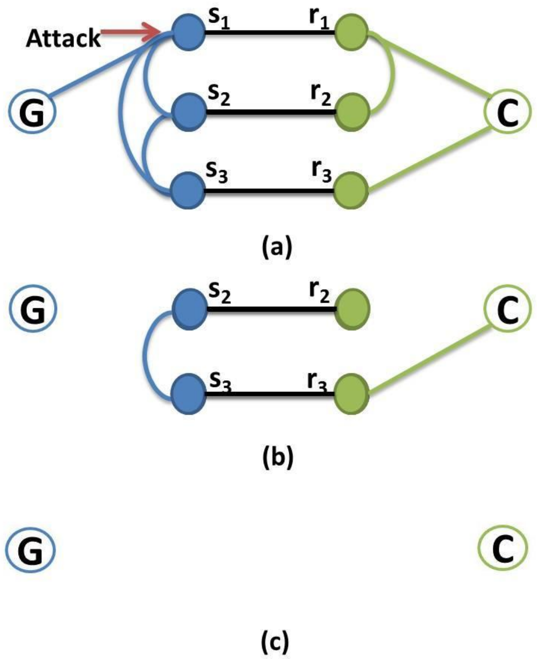

Figure 2.4: Representation of Interdependency and Cascading Failure in Power and Communication Network as Demonstrated by Parandehgheibi et al.

Keeping this failure model as their basis, the authors consider the problem of selection of the minimum number of non-autonomous nodes (substations, and routers), that need to be removed from the graph G, such that the resulting graph generated at the steady state contains no non-autonomous nodes. The authors term this problem as the *Node-MTFR* (minimum total failure removal) problem. They also identify another similar problem *Edge-MTFR*, that concentrates on the selection of the minimum number of edges of G such that the resulting graph generated at the steady state contains no non-autonomous node.

For solving these two problems the authors assume that the power network graph P, and the communication network graph C, are each star topology graphs. For the power network, the substations are directly connected to a generator without any connections between any other substations, i.e for all edges $(u, v) \in E_p$, node u

represents a substation, and node v represents a generator. Similarly for the communication network the router are directly connected to a control center without any connections between any other routers, i.e for all edges $(u, v) \in E_c$, node u represents a router node, and node v represents a control center node. The authors now proceed to analyze the problem from the perspective of a bipartite graph, where the nodes in the bipartite graph comprise of the substations of the power network, and routers of the communication network (the nodes representing generators, and control centers are ignored). The edges of this bipartite graph is the set of dependency relations represented by edge set E, of graph G. The authors analyze this problem from two interdependency perspectives, namely, unidirectional interdependency, and bi-directional interdependency.

For unidirectional dependency, the Node-MTFR problem is shown to be NP-complete by reduction from the Feedback Vertex Set problem, and an optimal solution is proposed by an integer linear program (ILP). A greedy approximation algorithm is also proposed for this problem and its solution is compared with the optimal solution obtained from the ILP. The authors also prove that Edge-MTFR problem for unidirectional interdependency is NP-complete by reduction from the Feedback Edge Set problem.

For bidirectional interdependency, the authors show that the Node-MTFR problem corresponds to a minimum vertex cover problem for bipartite graphs, and using Konig's Theorem, they show that this problem is equivalent to the maximum matching problem for bipartite graphs which has a known polynomial time solvable algorithm [28]. The authors also observe that for the Edge-MTFR problem with bidirectional interdependency all the edges of the bipartite graph must necessarily be removed, as any existing edge would denote the existence of operating non-autonomous nodes.

For the purpose of experimentation and simulation, the authors use the Italian communication and power network data obtained from [14]. To preserve the star topology configuration for the power and communication networks, only substations directly connected to the generators, and routers directly connected to control centers are considered. Unidirectional dependency between the substations and routers is established by assuming that a substation receives control signals from the nearest router, and a router receives power from the nearest substation. Using this setup the simulation is carried out to find the minimum number of nodes representing routers and substations that need to be removed, such that all non-autonomous nodes are removed from the graph (Node-MTFR). The experimental results showed that the north-western part of Italy is acutely vulnerable as removal of just three routers results in the failure of all substations and remaining routers.

A possible drawback to this model is that this model is able to represent dependencies that are in disjunctive form, for example, a sub-station survives as long it has a connection to a router. However, if there is a need to model a conjunctive dependency among network entities this model may not be adequate, for example, a scenario where a sub-station survives only when it is connected to two routers. In the real world, it is highly likely that entities in either the power or communication network have such conjunctive dependency amongst other entities, which this model may not be able to adequately represent. Another possible shortcoming of this model is the number of types of power, and communication entities that this model considers. For instance, in a real world communication network there may be communication entities such as cell towers whose survivability may have to be modeled very differently than the way routers are modeled. In the proposed model if support for additional entities are included that have different functional rules, it is not clear how this model will be able to accommodate them.

2.1.5 Castet et al. Interdependency Model

[23] develop a model for survivability analysis of networks with heterogeneous nodes (nodes that can perform more than one function), and apply their approach to space-based networks. The authors propose that heterogeneous networks can be modeled as interdependent multi-layer networks, thus enabling survivability analysis of these networks. They assert that in this approach, the multi-layer aspect captures the common functionalities across the different nodes (by construction of homogeneous sub-networks), and the interdependency aspect captures the physical characteristics of each node in the network.

In this paper the authors focuses on space-based networks (SBNs). In SBNs, each network entity (space-craft), may perform more than one function. SBN's operate by physically distributing functions in multiple orbiting space-crafts that are wirelessly connected to each other. The SBNs architecture allows the sharing of resources on-orbit, such as data processing, data storage, and downlinks among the network entities. In this study, Castet et al. attempt to assess their proposed approach of modeling heterogeneous networks as interdependent multi-layer networks on SBNs, and benchmark the survivability of a fractionated SBN architecture, against that of a traditional monolith spacecraft.

To represent the heterogeneous SBN as a multi-layer interdependent network the authors define the following terms: (a) *Super-Node*: A network entity that supports multiple functionalities, (b) *Node*: Component of a super-node that represents a single functionality of that super-node, (c) *Layer*: Set of nodes with the same functionality, (d) *Intra-Layer Link*: A link between two nodes in the same layer. The link can be directed (when one node is providing a resource and the other is receiving), or undirected (both provide, and receive resources), (e) *Networked Layer*: A network possessing intra-layer links, and (f) *Inter-Layer Link*: A directed link that captures the inter-dependency between functionalities (nodes) within a super-node. Specifically, this link implies the (directed) propagation of failure from one node to the other.

In their model two types of inter-layer links are considered that represent the two types of failure propagation possible in the model: (i) Inter-links for the *kill effect* failure propagation, defined by the propagation rule as follows: When a node fails, all nodes that have an incoming inter-link of this type from the failed node immediately fail, and (ii) Inter-links for the *precursor effect* failure propagation, defined by a conditional propagation rule as follows: When a node fails, and all the nodes with incoming intra-links to this failed node have also failed, all entities that have an incoming inter-link of this type from the failed node fails. This type of inter-link implicitly implies that as long as a super-node has access to a particular functionality, either from its own resources or from another super-node, all nodes in the super-node dependent on this functionality survive. Figure 2.5 demonstrates the propagation rules and represents a sample SBN as an interdependent multi-layer network N defined by $N(G_1, ..., G_L, E_k, E_p)$, where:

$$\begin{cases} L & is\ the\ number\ of\ layers\ each\ numbered\ sequentially\ from\ 1\ to\ L \\ & G_1, ..., G_L\ are\ the\ graphs\ on\ each\ layer: \\ & \quad \forall l \in [1, ..., L], G_l = (V_l, E_l)\ with: \\ & \qquad\qquad \begin{cases} V_l & is\ the\ set\ of\ n_l\ nodes\ in\ G_l \\ E_l & is\ the\ set\ of\ intra-layer\ links\ in\ G_l \end{cases} \\ E_k & is\ the\ set\ of\ inter-layer\ links\ representing\ the\ ''kill\ effect'' \\ E_p & is\ the\ set\ of\ inter-layer\ links\ representing\ the\ ''precursor\ effect'' \end{cases}$$

In Figure 2.5 the interdependent space based network consist of three layers represented by graphs $G_1 = (\{1, 2\}, \{(1, 2), (2, 1)\})$, $G_2 = (\{3, 4\}, \emptyset)$, $G_2 = (\{5\}, \emptyset)$. Edge set $E_k = \{(3, 1), (3, 5), (4, 2)\}$ and edge set $E_p = \{(1, 3), (1, 5), (2, 4)\}$. If node 3 fails, nodes 1 and 5 immediately fail (*kill effect*). If node 1 fails then nodes 3 and 5 don't fail unless node 2 also fails (*precursor effect*).

To analyze the survivability of an interdependent multi-layer network using the above network representation, and propagation rules, the authors carry out the following steps: (i) Generate the time to failure for each node and intra-layer link, (ii) propagate failures through inter-layer links for the kill effect, (iii) propagate failures through inter-layer links for the precursor effect, and (iv) combine all failure propagation effects to obtain the probability of failure of each node. Random times to failure for the nodes were generated using cumulative distribution functions representing the failure behavior of each node. Since links between two space-crafts (super-nodes) is established through a wireless unit, a two step process was followed for generating the times to failure for the intra-layer links: (i) times to failure of the wireless units on each spacecraft was generated using predetermined cumulative distribution functions, (ii) times to failures for each intra-layer link was generated by taking the minimum of the time to failures of the two associated wireless units.

For simulation and study, the authors apply their model into three different SBN scenarios. In their first scenario they consider three different space network architectures. The first architecture considered consists of a traditional monolith spacecraft with three subsystems (or layers), namely, *Telemetry Tracking and Command* (TTC), *supporting subsystems*, and *payload.* The second architecture consists of two space based networks, one of them a traditional monolith spacecraft, while the other spacecraft consists of two subsystems — *TTC* and *supporting subsystems.* The two

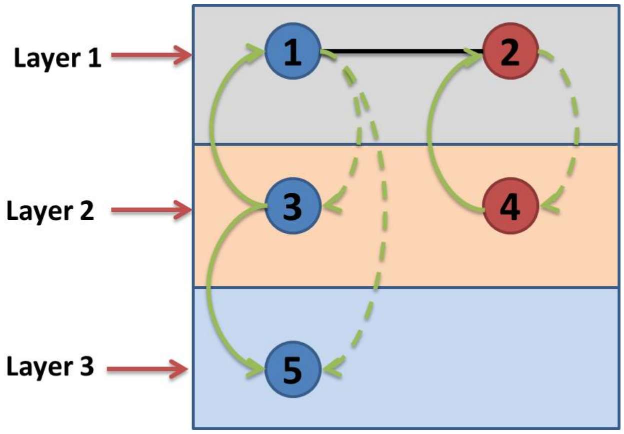

Figure 2.5: Representation of Interdependency and Cascading Failure in Power and Communication Network as Demonstrated by Castet et al.

spacecrafts shares their TTC subsystems, i.e. a TTC redundancy is introduced, through a wireless link. This architecture is shown in Figure 2.5 with layer 1,2 and 3 denoting subsystems TTC, supporting subsystems, and payload respectively. A third architecture is considered which is comprised of the monolith spacecraft, and two spacecrafts having two subsystems — *TTC* and *supporting subsystems*. These three spacecrafts share there TTC subsystems, i.e. there is a higher degree of TTC redundancy, through wireless links. Wireless links in the second and third spacecraft architecture are assumed to be perfect. The distribution of probability of unavailability (failure) of TTC subsystem with time, identified as a major spacecraft unreliability factor in [29], is obtained from [30]. The probability of unavailability of the payload subsystem over time, for the three spacecrafts is computed considering the failure of the TTC subsystem using a Monte Carlo Simulation. The simulation results showed that for a given time, increasing the redundancy of the TTC subsystems reduces the probability of unavailability of the payload. However, it was observed that the percentage of this reduction is not linear with the redundancy introduced.

The second scenario was aimed to study the impact of wireless link failure. A Weibull distribution is considered for probability of unavailability of wireless link failure with time. The parameters of Weibull distribution are set such that the wireless link has a probability of 0.5 to fail after 15 years. Simulations were carried out to compute the probability of unavailability of payload for the second architecture of the first application with the given wireless link failure distribution. The result is compared with the first and second architecture with perfect wireless link (the previous scenario). Compared with the first scenario, it was observed that for the second architecture the probability of unavailability of payload increases with time when wireless link failure is considered. At a given point in time, it surpasses the probability of unavailability of monolith spacecraft thus negating the effect of a TTC redundancy. The conclusion that can be drawn from these observations are that failure behavior of wireless links is a critical consideration to analyze the advantage of space based networks with TTC redundancy, over adoption of traditional monolith space crafts.

In the third scenario the authors consider a more complex space based network by including two new subsystems into the traditional monolith spacecraft. The new subsystems included are a Control Processor (CP) subsystem (the main computer of the spacecraft), and a Data Handling (DH) subsystem (handling exchange and storage of data). Another space craft is considered with all the subsystems as stated except the payload. These two spacecrafts share DH, TTC and CP subsystems, thus introducing redundancy. The resources are shared via wireless links. Hence the space based network represented by this architecture has 5 layers with 3 networked layer. The distribution of probability of unavailability of these subsystems with time is obtained from [30]. Assuming perfect wireless link, a Monte Carlo simulation is carried out to compute the probability of unavailability of payload with time. The simulation result is compared with traditional monolith spacecraft, and the second spacecraft architecture's (from the first scenario) payload failure distribution. It is observed that after 15 years this architecture reduces the risk of failure by 20.5% over the monolith spacecraft. This makes way to draw a conclusion that this architecture has greater improvement in reduction of failure over monolith spacecraft, than by only introducing TTC redundancy (as considered in first scenario).

2.1.6 Limitations of Current Modeling Approaches and Possible Solutions

As discussed, significant efforts have been made in the research community in the last few years to develop an appropriate model of interdependency between the entities of a multi-layer critical infrastructure network. Unfortunately, many of the proposed models are overly simplistic in nature and as such they fail to capture the complex interdependency that exists between power grid and communication networks. As noted in Section 2.1.1, the highly cited paper due to [8], assume that every node in one network can depend on one and only one node of the other network. Obviously, this assumption is not valid in an interdependent power-communication network that spans countries and continents. Even the authors in a follow up paper [13] recognize that the assumption may not be valid in the real world and a single node in one network may depend on more than one node in the other network and vice-versa. A node in one network may be functional ("alive") as long as one supporting node on the other network is functional.

Although this generalization can account for *disjunctive dependency* of a node in the A network (say a_i) on more than one node in the B network (say, b_j and b_k), implying that a_i may be "alive" as long as either b_i or b_j is alive, it cannot account for *conjunctive dependency* of the form when *both* b_j *and* b_k has to be alive in order for a_i to be alive. In a real network the dependency is likely to be even more complex involving both disjunctive and conjunctive components. For example, a_i may be alive if (i) b_j *and* b_k *and* b_l are alive, *or* (ii) b_m *and* b_n are alive, *or* (iii) b_p is alive. The graph based interdependency models proposed in the literature [14, 16, 15, 9, 23, 12] including [8, 13] cannot capture such complex interdependency between entities of multi-layer networks.

This dissertation as whole addresses this problem. It also describes the Implicative Interdependency Models which can capture dependencies between two infrastructures as well as dependencies that exist in a single infrastructure. In the course of this dissertation, some critical problems related to infrastructure systems which are beneficial in real world applications are described and solved.

Chapter 3

THE IMPLICATIVE INTER-DEPENDENCY MODEL

The need for a model to capture the complex intra and inter network dependencies is elaborated through a descriptive example of interdependent power and communication network. Consider the system shown in Figure 3.1 where the power network entities such as generators, transmission lines and substations are denoted by a_0 through a_{11} and communication entities such as GPS transmitters and satellites are denoted by b_0 through b_4. The Smart Control Center (SCC) is represented by the variable c_0 as it is a part of both the power and the communication network. For the SCC to be operational, it must receive electricity either from the generator via the different power grid entities, or from the battery. Similarly, the functioning of the generator will be affected if it fails to receive appropriate control signals from the SCC. The mutual dependency between the generator and the SCC can be expressed in terms of two implicative dependency relations — (i) $a_{11} \leftarrow b_4c_0$, (ii) $c_0 \leftarrow (b_0b_3(b_1+b_2))(a_0a_1+a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11})$. It may be noted that the SCC will not be operational if it does not receive electric power produced at the generating station and carried over the power grid entities to the SCC and its battery backup also fails. This dependency can be expressed by the implicative relation $c_0 \leftarrow a_0a_1+a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11}$ implying that c_0 will be operational (i) if entities a_0 and a_1 are operational, or (ii) if entities a_2 through a_{11} are operational. However, the SCC will also not be operational if it does not receive data from the communication system (IEDs, satellites, etc.). This dependency can be expressed by the relation $c_0 \leftarrow (b_0b_3(b_1+b_2))$. This implies that c_0 will be operational (i) if entities b_1 or b_2 is operational, and (ii) if entities b_0 and b_3 are operational. Combining the dependency of the SCC on the power grid and the communication network, the consolidated dependency relation can be expressed as $c_0 \leftarrow (b_0b_3(b_1+b_2))(a_0a_1+a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11})$. Likewise, the dependency relation for the generating station can be expressed as $a_{11} \leftarrow b_4c_0$, implying that the generating station will not be operational unless it receives appropriate signals from the SCC c_0, carried over wired or wireless link b_4. These two implicative relations demonstrate that dependency (or interdependency) is a complex combination of conjunctive and disjunctive terms. We term the model capturing this complex dependencies and interdependencies as *Implicative Interdependency Model*.

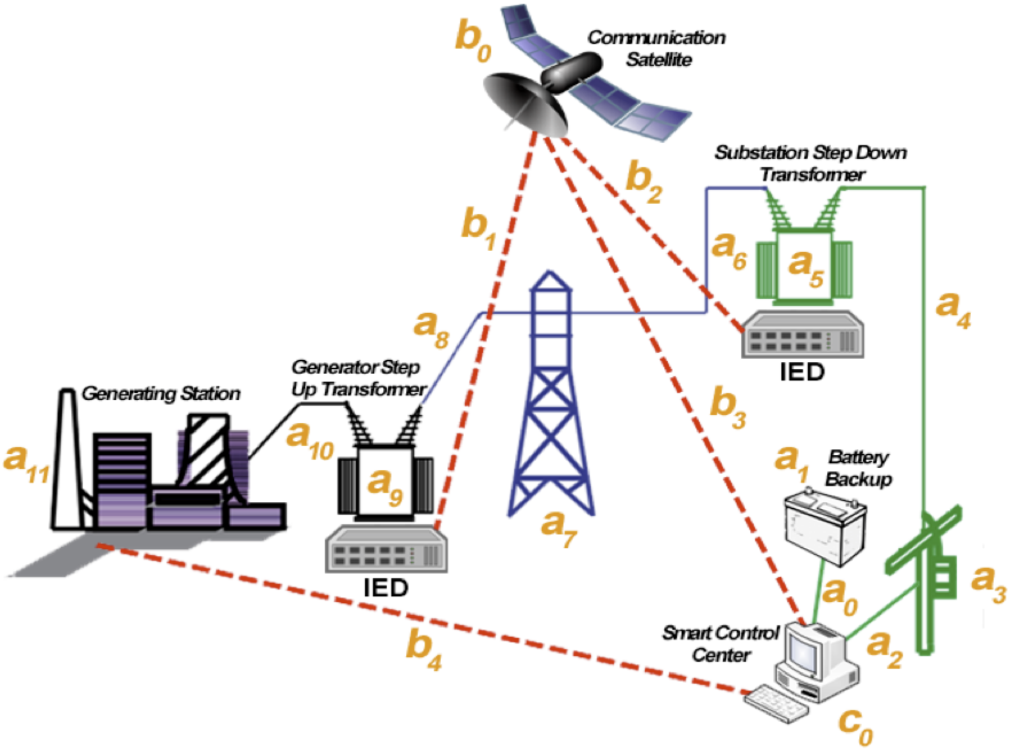

Figure 3.1: Example of Power - Communication Infrastructure Interdependency

In the IIM an intra-network or inter-network critical infrastructure system is represented by $\mathcal{I}(E, \mathcal{F}(E))$, where E is the set of entities and $\mathcal{F}(E)$ is the set of dependency relations. Throughout this dissertation, an intra-dependent critical infrastructure or interdependent critical infrastructure is termed as *system* denoted by $\mathcal{I}(E, \mathcal{F}(E))$. The dynamics of the model is explained through an example. Consider sets A and B (with $E = A \cup B$) representing entities in power and communication network (say) with $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3, b_4\}$ respectively. The function $\mathcal{F}(E)$ giving the set of dependency equations are provided in Table 3.1. In the given example, an IDR $b_3 \leftarrow a_2 + a_1a_3$ implies that entity b_3 is operational if entity a_2 *or* entity a_1 *and* a_3 are operational. In the IDRs each conjunction term e.g. a_1a_3 is referred to as *minterms*.

Power Network	**Comm. Network**
$a_1 \leftarrow b_2$	$b_1 \leftarrow a_1 + a_2$
$a_2 \leftarrow b_2$	$b_2 \leftarrow a_1a_2$
$a_3 \leftarrow b_4$	$b_3 \leftarrow a_2 + a_1a_3$
$--$	$b_4 \leftarrow a_3$

Table 3.1: IDRs for the Constructed Example

Initial failure of entities in $A \cup B$ would cause the failure to cascade until a steady state is reached. As noted earlier, the event of an entity failing after the initial failure is termed as *induced failure.* Failure in IIM proceeds in unit time steps with *initial failure* starting at time step $t = 0$. Each time step captures the effect of entities killed in all previous time steps. We demonstrate the cascading failure for the system outlined in Table 3.1 through an example. Consider the entities a_2 and a_3 fail at time step $t = 0$. Table 3.2 represents the cascade of failure in each subsequent time steps. In Table 3.2, for a given entity and time step, $'0'$ represents the entity is operational and $'1'$ non operational. In this example a steady state is reached at time step $t = 3$

Entities	Time Steps (t)						
	0	1	2	3	4	5	6
a_1	0	0	1	1	1	1	1
a_2	1	1	1	1	1	1	1
a_3	1	1	1	1	1	1	1
b_1	0	0	0	1	1	1	1
b_2	0	1	1	1	1	1	1
b_3	0	1	1	1	1	1	1
b_4	0	1	1	1	1	1	1

Table 3.2: Failure Cascade Propagation when Entities $\{a_2, a_3\}$ Fail at Time Step $t = 0$. A Value of 1 Denotes Entity Failure, and 0 Otherwise

when all entities are non operational. IIM also assumes that the dependent entities of all failed entities are killed immediately at the next time step. For example at time step $t = 1$ entities a_2, a_3, b_2, b_3 and b_4 are non operational. Due to the IDR $a_1 \leftarrow b_2$ entity a_1 is killed immediately at time step $t = 2$. At $t = 3$ the entity b_1 is killed due to the IDR $b_1 \leftarrow a_1 + a_2$ thus reaching the steady state.

As noted earlier the model captures the cascading failure that propagates through the entities on an event of *initial failure*. Consider $E = A \cup B$ with A and B representing entities in two separate critical infrastructures. The cascading failure process is shown diagrammatically in Figure 3.2 with sets $A_d^0 \subset A$ and $B_d^0 \subset B$ failing at $t = 0$. Accordingly, cascading failure in these systems can be represented as a *closed loop* control system shown in Figure 3.3. The steady state after an initial failure is analogous to the computation of *fixed point* of a function $\mathcal{G}(.)$ such that $\mathcal{G}(A_d^p \cup B_d^p) = A_d^p \cup B_d^p$, with steady state reached at $t = p$. It can be followed directly

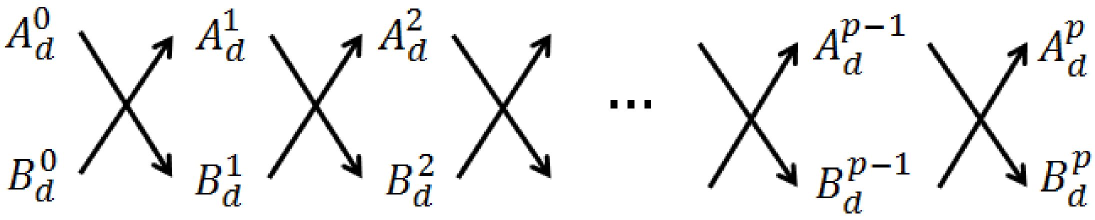

Figure 3.2: Cascading Failures Reach Steady State after p Time Steps

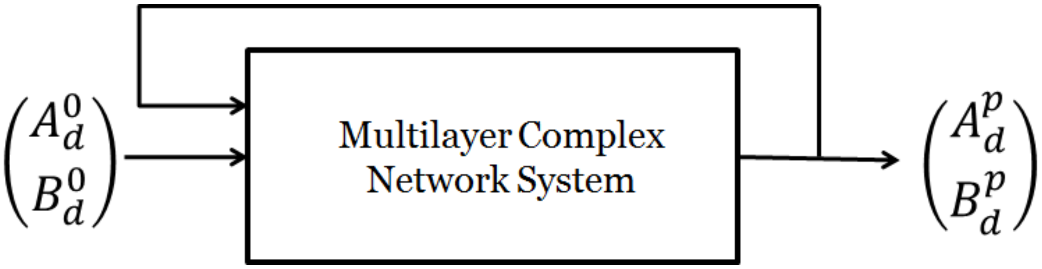
$\begin{pmatrix} A_d^0 \\ B_d^0 \end{pmatrix}$
Multilayer Complex Network System
$\begin{pmatrix} A_d^p \\ B_d^p \end{pmatrix}$

Figure 3.3: Cascading Failures as a Fixed Point System

that for a system with $|E| = m$, any initial failure would cause the system to reach a steady state within $m - 1$ time steps.

In the following section, methodologies for generating dependency equations for intra-dependent power network and inter dependent power and communication network are discussed. This methodologies are priliminary steps in deriving the dependency equations for a given intra/inter dependent critical infrastructure system. The dependency equation generation strategies are used to create test data sets to measure the efficacy of the solutions proposed for the problems that use IIM.

3.1 Generating IDRs

3.1.1 Generating Dependency Equations for Power Network

In this subsection, we describe a strategy to generate dependency equations of an intra-dependent power network. We restrict to load bus, generator bus, neutral bus and transmission line as the entities in the power network. For a given power network, AC power equations are solved to determine the direction of flow in the transmission lines. We use the power flow solver available in MatPower software for different bus systems [31]. For a given set of buses and transmission lines, the MatPower software uses load demand of the bus, impedance of the transmission lines etc. to solve the power flow and outputs the voltage of each bus in the system. We restrict to real power flow analysis. For a given solution, the real part of generation is taken as the power generated by a generator bus. Similarly, the real part of the load demand is taken as demand value of a load bus. For two buses e_1 and e_2 connected by a transmission line e_{12} the power flowing through the transmission line is calculated as $P_{12} = Real(V_1 * (\frac{V_1 - V - 2}{I_{12}})^*)$, where V_1 is the voltage at bus e_1, V_2 is the voltage at bus e_2, I_{12} is the impedance of the transmission line e_{12} and $(\frac{V_1 - V - 2}{I_{12}})^*$ denotes the complex conjugate of $(\frac{V_1 - V_2}{I_{12}})$. P_{12} is the real component of the power flowing in the transmission line e_{12}. Power flows from bus e_1 to e_2 if P_{12} is positive and from bus e_2 to e_1 otherwise.

Dependency Equations
$L_1 \leftarrow T_1 G_1$
$L_2 \leftarrow T_2 L_1 + T_7 N_2$
$L_3 \leftarrow T_3 L_1 + T_4 N_1$
$L_4 \leftarrow T_6 N_1 + T_8 N_2$
$N_1 \leftarrow T_5 G_3$
$N_2 \leftarrow T_9 G_2$

Table 3.3: IDRs of the Buses in Figure 3.4

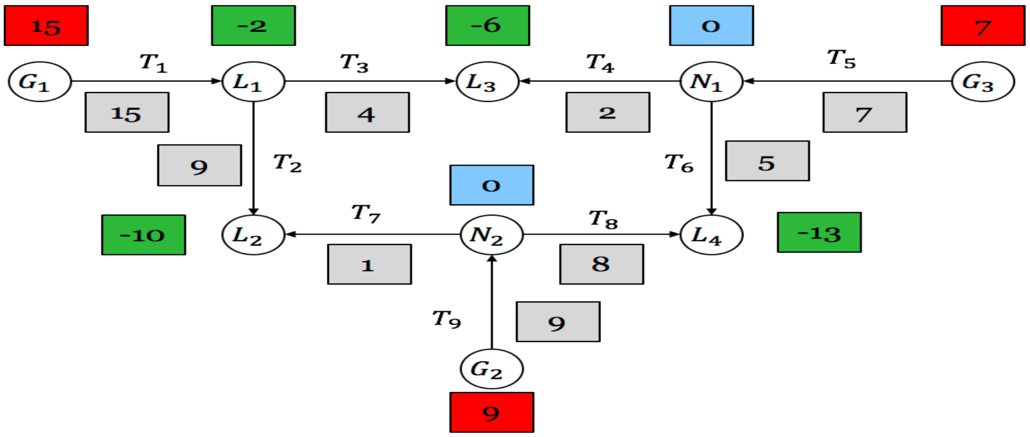

Figure 3.4: Example of Power Network Dependency

The generation of the dependency equation is explained through a nine bus system shown in Figure 3.4. The figure represents a system $\mathcal{I}(E, \mathcal{F}(E))$ with set E consisting of generator buses from G_1 through G_3, load buses L_1 through L_4, neutral buses $\{N_1, N_2\}$ and transmission lines T_1 through T_9. The values in the red blocks denote the amount of power a generator is generating, the green block being the load requirements and blue neutral (value of 0). The value in the grey blocks correspond to power flow in the transmission lines. The transmission lines don't have any IDR. The IDRs for a bus b_1 is constructed by the following — (a) let b_2, b_3 be the buses and b_{12} (between b_1 and b_2) and b_{13} between (b_1 and b_3) be the transmission lines for which power flows from these buses to b_1, (b) the dependency equation for the bus b_1 is constructed as conjunction of minterms of size 2 (consisting of the bus from which power is flowing and the transmission line) with each conjunction corresponding to bus that has power flowing to it. For this example the dependency equation $b_1 \leftarrow b_{12}b_2 + b_{13}b_3$ is created. Using this definition the dependency equations for the buses in Figure 3.4 are created and is shown in Table 3.3.

The following points are to be noted regarding the generation rule — (a) The transmission lines can only fail initially due to a man made attack or natural disaster. Hence it entails the underlying assumption that the transmission lines would have enough capacity to transmit any power that is required to flow in it, (b) The generator bus is also only susceptible to initial failure and is assumed to have a generation capacity that is enough to supply the power demanded by a instance of power flow, (c) Neutral and Load buses are prone to both initial and induced failure. For example consider the failure of transmission lines T_9 and T_1 at $t = 0$. Owing to this the load bus L_1 and neutral bus N_2 fails at $t = 2$. At $t = 3$ load bus L_2 fails due to the failure of buses L_1, N_2. It is to be noted that load bus L_3 does not fails as it still receives power from N_1 as transmission line T_4 is expected to have a capacity that can support a power flow equal to the demand of L_3.

Owing to the underlying assumptions in the the creation of dependency equations, there is a limitation to its applicability to real world problems. However, with respect to power network, creating dependency equations like the one discussed is a preliminary step. Further research is required to be done to have a more accurate abstract representation of the dependency equations that can have widespread applicability to real world problems. The purpose of this subsection is — (1) presenting a preliminary way the dependency equations can be generated for power network, (2) larger data sets that can be used to measure the performance of the optimal solution to the heuristic.

3.1.2 Generating Dependency Equations for Interdependent Power-Communication Network

In this subsection, we describe rules to generate dependency relations for interdependent power and communication network infrastructure as used in [18]. Real world

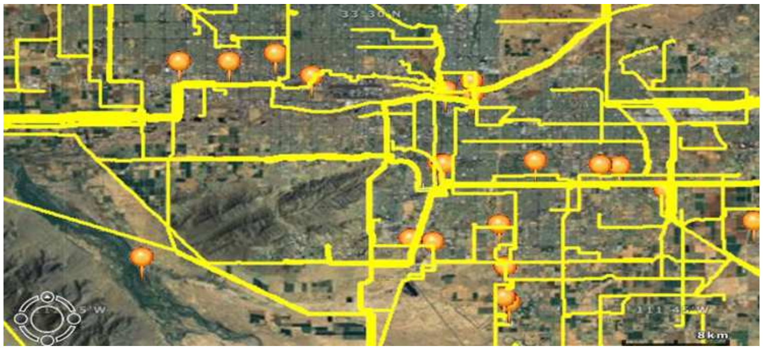

Figure 3.5: Snapshot of the Power Network

data of Maricopa County, Arizona , USA was taken. This county is one of the most densest populated region of Arizona with approximately 60% residents. Specifically, we wanted to measure the amount of resource required to protect entities in particular regions of the county when these regions have a set of entities failing initially. The data for power network was obtained from Platts (http://www.platts.com/) that contains 70 generator buses (including solar homes that generate minuscule unit of power) and 470 transmission lines. The communication network data was obtained from GeoTel (http://www.geo-tel.com/) consisting of 2, 690 cell towers, 7, 100 fiber-lit buildings and 42, 723 fiber links. Figures 3.5 and 3.6 displays the snapshot of power network and communication network for a particular region of Maricopa county. In Figure 3.5 the orange dots represent the generator buses and continuous yellow lines represent the transmission lines. In Figure 3.6 fiber-lit buildings are represented by pink dots, cell towers by orange dots and fiber links by continuous green lines.

The *load* of the power network are assumed to be cell towers and fiber-lit buildings. There exist other entities that draws electrical power. Since it is not relevant for the comparative analysis of the heuristic and the ILP such entities are ignored. The interdependent power-communication system is represented mathematically as $\mathcal{I}(E, \mathcal{F}(E))$ with $E = A \cup B$. A and B consist of the entities in the power net-

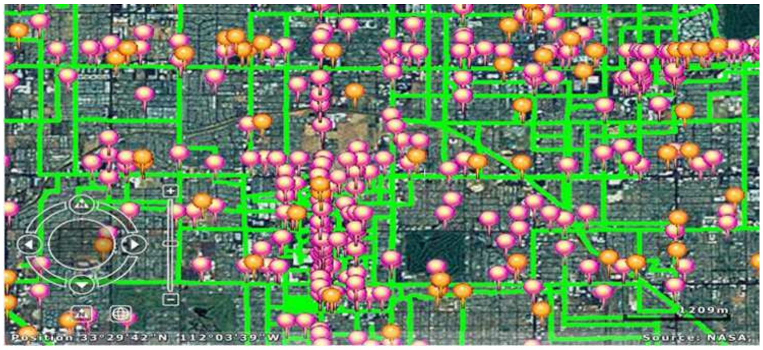

Figure 3.6: Snapshot of the Communication Network

work and communication network respectively. With respect to this data the power network consist of three type of entities — generating stations, load (which are cell towers and fiber-lit buildings) and transmission lines (denoted by a_1, a_2, a_3 respectively). The communication network comprises of the following type of entities — cell towers, fiber-lit buildings and fiber links (denoted by b_1, b_2, b_3 respectively). It is to be noted that the fiber-lit buildings and cell towers are considered as both power network entities as well as communication network entities. From the raw data the dependency equations are constructed using the following rules.

Rules: We take into consideration that an entity in the power network is dependent on a set of entities in the communication network for either being operational and vice-versa. To keep things uncomplicated, we consider the dependency equations with at most two minterms. For the same reason we consider the size of each minterm is at most two.

Generators ($a_{1,i}, 1 \leq i \leq p$, where p is the total number of generators): We assume that every generator ($a_{1.i}$) is, i) dependent on the closest Cell Tower ($b_{1,j}$), or, ii) closest Fiber-lit building ($b_{2,k}$) and the corresponding Fiber link ($b_{3,l}$) connecting $b_{2,k}$ and $a_{1,i}$. Hence, we have $a_{1,i} \leftarrow b_{1,j} + b_{2,k} \times b_{3,l}$.

Load ($a_{2,i}, 1 \leq i \leq q$, where q is the total number of loads): The power network loads do not depend on any entities in communication network

Transmission Lines ($a_{3,i}, 1 \leq i \leq r$, where r is the total number of transmission lines): The transmission lines in the power network do not depend on any entities in communication network.

Cell Towers ($b_{1,i}, 1 \leq i \leq s$, where s is the total number of cell towers): The Cell Towers depend on two components, i) the closest pair of generators, and, ii) corresponding transmission line, connecting the generator to the cell tower. Thus we have $b_{1,i} \leftarrow a_{1,j} \times a_{3,k} + a_{1,j'} \times a_{3,k'}$.

Fiber-lit Buildings ($b_{2,i}, 1 \leq i \leq t$, where t is the total number of fiber-lit buildings): The Fiber-lit Buildings depend on two components, i) the closest pair of generators, and, ii) corresponding transmission line, connecting the generator to the fiber-lit buildings. Thus we have $b_{2,i} \leftarrow a_{1,j} \times a_{3,k} + a_{1,j'} \times a_{3,k'}$.

Fiber Links ($b_{3,i}, 1 \leq i \leq u$, where u is the total number of fiber links): The Fiber Links aren't dependent on any power network entity. These links require power only for the amplifiers connected to them. The amplifiers are required if the length of the fiber link is above a certain threshold. We consider only those fiber links which are 'quite long', need power. The fiber links depend on the closest pair of generators and the transmission lines connecting the generators to the fiber link under consideration. Thus we have $b_{3,i} \leftarrow a_{1,j} \times a_{3,k} + a_{1,j'} \times a_{3,k'}$. We do not consider that these fiber links need any power as we cannot determine the length of the fiber links or the exact threshold value due to the lack of data.

Chapter 4

THE ENTITY HARDENING PROBLEM

For an existing critical infrastructure system, an operator would have the capability to measure the extent of failure when a certain set of entities fail initially. Consider a scenario where an operator identifies a set of critical entities which when failed initially would cause the maximum damage. In an ideal case, there would be enough resources available to support those critical entities from initial failure. However, if the availability of resources is a constraint, then an operator might have to choose entities which when supported would minimize the damage. We define the entities to support as the entities to harden and the problem as the *Entity Hardening Problem*. An entity x_i when hardened is resistant to both *initial* and *induced* failure (failing of entities in the cascading process after the initial failure). In the physical world, an entity can be hardened with respect to cyber attacks (say) by having a strong firewall. Similarly some entities can be hardened by — (a) strengthening their physical structures for protection from natural disaster, (b) placing redundant entity a' for an entity a which can operate when a fails, (c) increasing physical limits of the entity (maximum power flow capacity of the transmission line, maximum generation capacity of a generator bus). There exist multiple such ways to harden an entity from different kind of failures. Even though there may be circumstances under which an entity cannot be hardened, we relax such possibilities and assume that there always exist a way to harden a given entity. Hardening entities can prevent cascading failures caused by some initial failure. Thus this results in protecting a set of entities including the hardened entities from an initial failure trigger. Using these definitions the *Entity Hardening Problem* finds a set of k entities that should be hardened (with k being the resource constraint) in an intra-network or inter-network critical infrastructure system that protects the maximum number of entities from failure when a set of K entities fail initially.

4.1 Problem Formulation

Before stating the problem formally, a brief understanding of entity hardening is provided. Consider the system with set of dependency relations given by Table 3.1 of Chapter 3. With an initial failure of entities a_2, a_3 the subsequent cascading failures is shown in Table 3.2 which fails all the entities in the system. We note three instances where entities a_1, a_2 and a_3 are hardened separately with a_2, a_3 failing initially. The failure cascade propagation when a_1, a_2 and a_3 are hardened are shown in Tables 4.1, 4.2, and 4.3 respectively. In the tables the cascading failure is shown till $t = 3$ because with initial failure of entities a_2, a_3 the cascade propagation stops at $t = 3$ as seen in Table 3.2. Hardening entity a_1 protect entities a_1, b_1 from failure. Similarly, when a_2 is hardened it protect a_1, a_2, b_1, b_2, b_3 and hardening a_3 protect entities a_3, b_4. If the hardening budget is 1 the operator would clearly harden the entity a_2 as it protects the maximum number of entities from failure. We now describe the entity hardening problem formally.

The Entity Hardening (ENH) problem

INSTANCE: Given:

(i) A system $\mathcal{I}(E, \mathcal{F}(E))$, where the set E represent the set of entities, and $\mathcal{F}(E)$ the set of IDRs.

(ii) The set of K initially failing entities E', where $E' \subseteq E$

(iii) Two positive integers $k, k < K$ and E_F.

Entities	Time Steps (t)			
	0	1	2	3
a_1	0	0	0	0
a_2	1	1	1	1
a_3	1	1	1	1
b_1	0	0	0	0
b_2	0	1	1	1
b_3	0	1	1	1
b_4	0	1	1	1

Table 4.1: Failure Cascade Propagation when Entities $\{a_2, a_3\}$ Fail at Time Step $t = 0$ and a_1 is Hardened

Entities	Time Steps (t)			
	0	1	2	3
a_1	0	0	0	0
a_2	0	0	0	0
a_3	1	1	1	1
b_1	0	0	0	0
b_2	0	0	0	0
b_3	0	0	0	0
b_4	0	1	1	1

Table 4.2: Failure Cascade Propagation when Entities $\{a_2, a_3\}$ Fail at Time Step $t = 0$ and a_2 is Hardened

Entities	Time Steps (t)			
	0	1	2	3
a_1	0	0	1	1
a_2	1	1	1	1
a_3	0	0	0	0
b_1	0	0	0	1
b_2	0	1	1	1
b_3	0	1	1	1
b_4	0	0	0	0

Table 4.3: Failure Cascade Propagation when Entities $\{a_2, a_3\}$ Fail at Time Step $t = 0$ and a_3 is Hardened

DECISION VERSION: Is there a set of entities $\mathcal{H} = E'', E'' \subseteq E, |\mathcal{H}| \leq k$, such that hardening $\mathcal{H}$ entities results in no more than E_F entities to fail after entities in E' fail at time step $t = 0$.

OPTIMIZATION VERSION: Find a set of k entities to harden which would maximize the number of protected entities with entities in E' failing initially.

Definition: *$KillSet(S)$: For an initial failure of set S, the set of entities that fail due to induced failure in the cascading process including the entities in set S is denoted by $KillSet(S)$.*

The following points are to be noted regarding the ENH problem — (a) the condition $k < K$ is assumed as with $k \geq K$ hardening the K initially failing entities would ensure that there are no induced and initial failure. (b) with E' entities failing initially, the entities to be harden are to be selected from $KillSet(E')$. Hardening entities outside $KillSet(E')$ would not result in protection of any non-hardened entity.

4.2 Computational Complexity Analysis

The computational complexity of the ENH problem is provided in this section. The problem is proved to be NP-complete. Additionally, approximate and polynomial solutions to few subcases are provided. The subcases impose restrictions on the IDRs and the solutions can be applied to systems whose dependency equations fall within the definition of the restriction. We prove that the ENH problem is NP-complete in Theorem 1. Using the results of Theorem 1 an in-approximability bound of the problem is provided in Theorem 2.

Theorem 1. *The ENH Problem is NP Complete*

Proof. The Entity Hardening problem is proved to be NP complete by giving a reduction from the Densest p-Subhypergraph problem [32], a known NP-complete problem. An instance of the Densest p-Subhypergraph problem includes a hypergraph $G = (V, E_V)$, a parameter p and a parameter M. The problem asks the question whether there exists a set of vertices $|V'| \subseteq V$ and $|V'| \leq p$ such that the subgraph induced with this set of vertices has at least M hyperedges that are completely covered. From an instance of the Densest p-Subhypergraph problem we create an instance of the ENH problem in the following way. Consider a system $\mathcal{I}(E, \mathcal{F}(E))$ with $E = A \cup B$, where A and B are entities of two separate critical infrastructures dependent on each other. For each vertex v_i and each hyperedge e_j entities b_i and a_j are added to the set B and A respectively. For each hyperedge e_j with $e_j = \{v_m, v_n, v_q\}$ (say) an IDR of form $a_j \leftarrow b_m b_n b_q$ is created. It is assumed that the value of K is set of $|V|$. The values of k and E_F are set to p and $|V| + |E_V| - p - M$ (where $|A| = |V|$ and $|B| = |E|$) respectively.

In the constructed instance only entities of set A are dependent on entities of set B. Additionally the dependency for an entity a_i consists of conjunction of entities in set B. Hence for an entity $a_i \in A$ to fail, either it itself has to fail initially or any one of the entity that a_i depends on has to fail. It is to be noted that the entities in set B has no induced failure i.e., there is no cascade. Following from this assertion, with $K = |V'|$, failing entities in B would fail all entities in set $A \cup B$. For this created instance E' is set to B'

If an entity in set A is hardened then it would have no effect in failure prevention of any other entities. Whereas hardening an entity $b_m \in B$ might result in failure prevention of an entity $a_i \in A$ with IDR $a_j \leftarrow b_m b_n b_q$ provided that entities b_n, b_q are also defended. With $k = p$ (and $K \leq |V| = |B|$) it can be ensured that entities to be defended are from set B'.

To prove the theorem, consider that there is a solution to the Densest p-Subhypergraph problem. Then there exist p vertices which induces a subgraph which has at least M hyperedges. Hardening the entities $b_i \in B'$ for each vertex v_i in the solution of the Densest p-Subhypergraph problem would then ensure that at least M entities in set A are protected from failure. This is because the entities in set A for which the failure is prevented corresponds to the hyperedges in the induced subgraph. Thus the number of entities that fail after hardening p entities is at most $|V| + |E_V| - p - M$, solving the ENH problem. Now consider that there is a solution to the ENH problem. As previously stated, the entities to be hardened will always be from set B'. So defending p entities from set B' would result in failure prevention of at least M entities in set A such that $E_F \leq |V| + |E_V| - p - M$. Hence, the vertex induced subgraph would have at least M hyperedges completely covered when vertices corresponding to the entities hardened are included in the solution of the Densest p-Subhypergraph problem. Hence proved. □

Theorem 2. *For a system $\mathcal{I}(E, \mathcal{F}(E))$ with $n = |E|$ and $\mathcal{F}(E)$ having IDRs of form in the created instance of Theorem 1, the ENH problem is hard to approximate within a factor of $\frac{1}{2^{log(n)^{\lambda}}}$ for some $\lambda > 0$.*

Proof. The ENH problem with IDRs of form in the created instance of Theorem 1 is a special case of the densest p-subhypergraph problem. In [32] the densest $p-$subhypergraph problem is proved to be inapproximable within a factor of $\frac{1}{2^{log(n)^{\lambda}}}$ $(\lambda > 0)$. The same result applied to the ENH problem as well. Hence the theorem follows. □

4.2.1 Restricted Case I: Problem Instance with One Minterm of Size One

The IDRs of this restricted case have a single minterm of size 1. This can be represented as $e_i \leftarrow e_j$, where e_i and e_j are entities of a system $\mathcal{I}(E, \mathcal{F}(E))$. Algorithm 1 solves the ENH problem with this restriction optimally in polynomial time utilizing the notion of *Kill Set* defined in Section 4.1 with proof of optimality given in Theorem 3.

Theorem 3. *Algorithm 1 solves the Entity Hardening problem for the Restricted Case I optimally in polynomial time.*

Proof. It is shown in [18] that the kill set for all entities in a system can be computed in $\mathcal{O}(n^3)$ where $n = |E|$. Thus computing the kill sets of K entities would have a time complexity of $\mathcal{O}(Kn^2)$. Each update in line 8 would take $\mathcal{O}(n)$ time and hence the total computation of the inner for loop can be done in $\mathcal{O}(Kn)$. The outer for

Algorithm 1: Entity Hardening Algorithm for systems with Restricted Case I type of dependencies

```
   Data: A system $\mathcal{I}(E, \mathcal{F}(E))$, set of $K$ entities failing initially $E', E' \subseteq E$, hardening
         budget $k$
   Result: Set of hardened entities $\mathcal{H}$
1  begin
2      For each entity $e_i \in E'$ compute the set of kill sets and store it in a set
         $\mathcal{C} = \{C_{e_1}, C_{e_2}, ..., C_{e_K}\}$, where $C_{e_i} = KillSet(e_i)$ ;
3      Set $\mathcal{H} = \emptyset$ ;
4      for (i=1; i $\leq \mathcal{K}$; i++) do
5          Choose the set $C_{e_k}$ having the highest cardinality from $\mathcal{C}$ ;
6          Update $\mathcal{C} \leftarrow \mathcal{C} \setminus C_{e_k}$ ;
7          for $C_{e_j} \in \mathcal{C}$ do
8              Update $C_{e_j} \leftarrow C_{e_j} \setminus C_{e_k}$ ;
9          Update $\mathcal{H} \leftarrow \mathcal{H} \cup \{e_k\}$;
10         If all Kill Sets are empty then break ;
11     return $\mathcal{H}$
```

loop iterates for K times thus the time complexity of lines $4 - 9$ is $\mathcal{O}(K^2n)$. Hence Algorithm 1 runs in $\mathcal{O}(Kn^2)$.

For two kill sets C_{e_i} and C_{e_j}, it can be shown that either $C_{e_i} \cap C_{e_j} = \emptyset$ or $C_{e_i} \cap C_{e_j} = C_{e_i}$ or $C_{e_i} \cap C_{e_j} = C_{e_j}$ [18]. Using this assertion the set E' can be partitioned into disjoint subsets $E_{X_1}, E_{X_2}, .., E_{X_m}$ where kill sets of two entities e_a, e_bhave no elements in common with $e_a \in E_{X_i}$ and $e_b \in E_{X_j}$ and $i \neq j$. Additionally, for any given subset of entities E_{X_z} there exist an entity $e_k \in E_{X_z}$ whose kill set is a super set of kill sets of all other entities in E_{X_z}. Thus each of the disjoint subset has an entity whose kill set is the super set among all other entities in that subset. Algorithm 1 chose such an entity in line 5 for every iteration and updates in line 8 would make the kill set of all the remaining entities in the partition to be empty and hence would not be hardened in future iterations. Clearly choosing these entities would globally maximize the total number of protected entities from failure. Hence the Algorithm 1 is proved to be optimal. □

4.2.2 Restricted Case II: Problem Instance with an Arbitrary Number of Minterms of Size One

The IDRs of this restricted case have arbitrary number of minterm of size 1. This can be represented as $e_i \leftarrow \sum_{q=1}^{p} e_q$, where e_i and e_q are entities of a system $\mathcal{I}(E, \mathcal{F}(E))$ and the number of minterms are p. The ENH problem with respect to this restricted case is NP-complete and is proved in Theorem 4. We provide an approximation bound for this restricted case of the problem in Theorem 6 using the results of Theorem 4. The approximation bound uses the notion of *Protection Set*. The Protection Set of an entity can be computed in $\mathcal{O}((n)^2)$ where $n = |E|$ and m are number of minterms.

Definition: *For an entity $e_i \in E$ the Protection Set is defined as the entities that would be prevented from failure by hardening the entity e_i when all entities in E' fail initially. This is represented as $P(x_i|E')$.*

Theorem 4. *The ENH problem for Restricted Case II is NP Complete*

Proof. The ENH problem for case III is proved to be NP complete by giving a reduction from the Set Cover Problem. An instance of the Set Cover problem is given by a set $S = \{x_1, x_2, ..., x_n\}$ of elements, a set of subsets $\mathcal{S} = \{S_1, S_2, ..., S_m\}$ where $S_i \subseteq S$ and a positive integer M. The decision version of the problems finds whether there exist at most M subsets from set $\mathcal{S}$ whose union would result in the set S. From an instance of the set cover problem we create an instance of the ENH problem in the following way. Consider a system $\mathcal{I}(E, \mathcal{F}(E))$ with $E = A \cup B$, where A and B are entities of two separate critical infrastructures dependent on each other. For each element x_i in set S we add an entity a_i in set A. For each subset S_i in set $\mathcal{S}$ we add an entity b_i in set B. For all subsets in $\mathcal{S}$, say S_p, S_m, S_n, which has the element x_i an IDR of form $a_i \leftarrow b_m + b_n + b_l$ is added to $\mathcal{F}(E)$. The values of positive integers k and E_F are set to M and $m - M$ respectively. It is assumed that the value of $K = m$ and $E' = B$.

The constructed instance ensures that the entities to be hardened are from set B. This is because hardening an entity $a_i \in A$ would only result in prevention of its own failure whereas hardening an entity $b_j \in B$ would result in failure prevention of its own and all other entities in set A for which it appears in its IDR.

Consider there exists a solution to the Set Cover problem. Then there exist M subsets whose union results in the set S. Hardening the entities in set B corresponding to the subsets selected would ensure that all entities in set A are prevented from failure. This is because for the dependency of each entity $a_i \in A$ there exist at least one entity (in set B) that is hardened. Hence the number of entities that fails after hardening is $m - M$ which is equal to E_F, thus solving the ENH problem. Now, consider that there is a solution to the ENH problem. As discussed above the entities to be hardened should be from set B'. To achieve $E_F = m - M$ with $k = M$, no entities in the set A must fail. Hence for each entity $a_i \in A$ at least one entity in set B that appears in its IDR has to be hardened. Thus, it directly follows that the union of subsets in set $\mathcal{S}$ is equal to the set S, solving the Set Cover Problem. Hence proved. $\square$

Theorem 5. *For two entities $e_i, e_j \in A \cup B$, $P(e_i|E') \cup P(e_j|AE') = P(e_i, e_j|E')$ when IDRs are in form of Restricted Case II.*

Proof. Assume that defending two entities e_i and e_j would result in preventing failure of $P(e_i, e_j|E')$ entities with $|P(e_i|E') \cup P(e_j|E')| < |P(e_i, e_j|E')|$. Then there exist at least one entity $e_p \notin P(e_i|E') \cup P(e_j|E')$ such that it's failure is prevented only if e_i and e_j are protected together. So two entities e_m and e_n (with $e_m \in P(e_i|E')$ and $e_n \in P(e_j|E')$ or vice versa) have to be present in the IDR of e_p. As the IDRs are of restricted Case II so if any one of e_m or e_n is protected then e_p is protected, hence a contradiction. On the other way round $P(e_i, e_j|E')$ contains all entities which would be prevented from failure if e_i or e_j is defended alone. So it directly follows that $|P(e_i|E') \cup P(e_j|E')| > |P(e_i, e_j|E')|$ is not possible. Hence the theorem holds. □

Theorem 6. *There exists an $1 - \frac{1}{e}$ approximation algorithm that approximates the ENH problem for Restricted Case II.*

Proof. The approximation algorithm is constructed by reducing the problem for this restricted case to *Maximum Coverage* problem. An instance of the maximum coverage problem consists of a set $S = \{x_1, x_2, ..., x_n\}$, a set $\mathcal{S} = \{S_1, S_2, ..., S_m\}$ where $S_i \subseteq S$ and a positive integer M. The objective of the problem is to find a set $S' \subseteq S$ and $|S'| \leq M$ such that $\cup_{S_i \in \mathcal{S}} S_i$ is maximized. Consider a system $\mathcal{I}(E, \mathcal{F}(E))$ with $E = A \cup B$, where A and B are entities of two separate critical infrastructures dependent on each other. For a given initial failure set $E' = A' \cup B'$ with $|A'| + |B'| \leq K$, let $P(e_i|A' \cup B')$ denote the protection set for each entity $e_i \in A \cup B$. We construct a set $S = A \cup B$ and for each entity e_i a set $S_{e_i} \subseteq S$ such that $S_{e_i} = P(e_i|A' \cup B')$. Each set S_{e_i} is added as an element of a set $\mathcal{S}$. The conversion of the problem to Maximum Coverage problem can be done in polynomial time. By Theorem 5 defending a set of entities $X \subseteq S$ would result in failure prevention of $\cup_{e_i \in X} S_{x_i}$ entities. Hence, with the constructed sets S and $\mathcal{S}$ and a positive integer M (with $M = k$) finding the Maximum Coverage would ensure the failure protection of maximum number of entities in $A \cup B$. This is same as the ENH problem of Restricted Case II. As there exists an $1 - \frac{1}{e}$ approximation algorithm for the Maximum Coverage problem hence the same algorithm can be used to solve this restricted case of the ENH problem using this transformation. $\square$

4.3 Optimal and Heuristic Solution to the Problem

Owing to the problems being NP-complete, we provide an optimal solutions to the problem by formulating Integer Linear Program (ILP). We also provide sub optimal heuristic that runs in polynomial time.

4.3.1 Optimal Solution using Integer Linear Programming

We propose an Integer Linear Program (ILP) that solves the ENH problem optimally. For a system $\mathcal{I}(E, \mathcal{F}(E))$ let $G = \{g_1, g_2, ..., g_n\}$ be variables denoting entities in set E. Given an integer K, G is a array of K 1's and $n - K$ 0's where $g_i = 1$ if the entity $e_i \in E$ fails at $t = 0$ and $g_i = 0$ if the the entity is operational at $t = 0$. Thus the array G gives the set of K entities failing initially. Additionally for each entity $e_j \in E$ a set of variables x_{jd} with $0 \leq d \leq n - 1$ and $d \in I^{+} \cup \{0\}$ are created. The value of $x_{jd} = 1$ denotes that the entity x_j is in failed state at $t = d$ and $x_{jd} = 0$ denotes it is operational. As noted earlier for $|E| = n$ the cascade can proceed till $n - 1$ so the range of d is $[0, n - 1]$. Using these definitions the objective of the ENH problem is as follows —

$$min\Big(\sum_{i=1}^{n} x_{i(n-1)}\Big) \tag{4.1}$$

The constraints guiding the problem are as follows:

Constraint Set 1: $\sum_{i=1}^{n} q_{x_i} \leq k$, with $q_{x_i} \in [0,1]$. If an entity x_i is hardened then $q_{x_i} = 1$ and 0 otherwise.

Constraint Set 2: $x_{i0} \geq g_i - q_{x_i}$. This constraint implies that only if an entity is not defended and $g_i = 1$ then the entity will fail at the initial time step.

Constraint Set 3: $x_{id} \geq x_{i(d-1)}, \forall d, 1 \leq d \leq n-1$, in order to ensure that for an entity which fails in a particular time step would remain in failed state at all subsequent time steps.

Constraint Set 4: Modeling of constraints to capture the cascade propagation for IIM is similar to the constraints established in [18] with modifications to capture the hardening process. A brief overview of this constraint is provided here. Consider an IDR $e_i \leftarrow e_j e_p e_l + e_m e_n + e_q$. The following steps are enumerated to depict the cascade propagation with respect to this constraint:

Step 1: Replace all minterms of size greater than one with a variable. In the example provided we have the transformed minterm as $e_i \leftarrow c_1 + c_2 + e_q$ with $c_1 \leftarrow e_j e_p e_l$ and $c_2 \leftarrow e_m e_n$ ($c_1, c_2 \in \{0,1\}$) as the new IDRs.

Step 2: For each variable c, a constraint is added to capture the cascade propagation. Let N be the number of entities in the minterm on which c is dependent. In the example for the variable c_1 with IDR $c_1 \leftarrow e_j e_p e_l$, constraints $c_{1d} \geq \frac{x_{j(d-1)}+x_{p(d-1)}+x_{l(d-1)}}{N} \forall d \in [1, n-1]$ are introduced ($N = 3$ in this case). If IDR of an entity is already in form of a single minterm of arbitrary size, i.e.,$e_i \leftarrow e_j e_p e_l$ (say) then constraints

$x_{id} \geq \frac{x_{j(d-1)}+x_{p(d-1)}+x_{l(d-1)}}{N} - q_{x_i}$ and $x_{id} \leq x_{j(d-1)} + x_{p(d-1)} + x_{l(d-1)} \forall d \in [1, n-1]$ are introduced (with $N = 3$). These constraints satisfies that if the entity e_i is hardened initially then it is not dead at any time step.

Step 3: Let M be the number of minterms in the transformed IDR as described in Step 1. In the given example with IDR $e_i \leftarrow c_1 + c_2 + e_q$ constraints of form $x_{id} \geq c_{1(d-1)} + c_{2(d-1)} + x_{q(d-1)} - (M-1) - q_{x_i}$ and $x_{id} \leq \frac{c_{1(d-1)}+c_{2(d-1)}+x_{q(d-1)}}{M} \forall d \in [0, 1]$ are introduced. These constraints ensures that even if all the minterms of e_i has at least one entity in dead state then it will be alive if the entity is hardened initially.

With objective (4.1) along with the constraints minimize the number of entities failed at the end of the cascading failure with a hardening budget of k and K entities failing initially. The ILP gives an optimal solution to the ENH problem, however its run time is non-polynomial.

4.3.2 Heuristic Solution

In this subsection we provide a greedy heuristic solution to the Entity Hardening problem. For a given system $\mathcal{I}(E, \mathcal{F}(E))$ with set of entities $E'(|E'| = K)$ failing initially and hardening budget k, a heuristic is developed based on the following two metrics — (a) *Protection Set* as defined in Section 6.1, (b) *Cumulative Fractional Minterm Hit Value (CFMHV).*

Definition: The Fractional Minterm Hit Value of an entity $e_j \in E$ in a system $\mathcal{I}(E, \mathcal{F}(E))$ is denoted as $FMHV(e_j, X)$. It is calculated as $FMHV(e_j, X) = \sum_{i=1}^{m} \frac{1}{|s_i|}$. In the formulation m are the minterms in which e_j appears over all IDRs except for the IDRs of entities in set X. The parameter s_i denotes i^{th} such minterm. If entity e_j is hardened (or protected from failure) then the computed value provides an estimate of the future impact on protection of other non operational entities.

Definition: The Cumulative Fractional Minterm Hit Value of an entity $e_j \in E$ is denoted as $CFMHV(e_j)$ where $CFMHV(e_j) = \sum_{\forall x_i \in PS(e_j|E')} FMHV(x_i, PS(x_i|E'))$. This gives a measure of the future impact on protecting non functional entities when the entity e_j is hardened and entities $PS(e_j|E')$ are protected from failure.

Using these definitions a heuristic is formulated in Algorithm 2. For each iteration of the while loop in the algorithm, the entity having highest cardinality of the set $PS(x_i|E')$ is hardened. This ensures that at each step the number of entities protected is maximized. In case of a tie, the entity having highest Cumulative Fractional Minterm Hit Value among the set of tied entities is selected. This causes the selection of an entity that has the potential to protect maximum number of entities in subsequent iterations. Thus, the heuristic greedily maximizes the number of entities protected when an entity is hardened at the current iteration with metric to measure its impact of protecting other non operational entities in future iterations. Algorithm 2 runs in polynomial time, more specifically the time complexity is $\mathcal{O}(kn(n+m)^2)$ (where $n = |E|$ and $m =$ Number of minterms in $\mathcal{F}(E)$).

4.4 Experimental Results

A comparative study of the ILP and heuristic solution for the ENH problems is done in this section. A machine with intel i5 processor and 8 GB of RAM was used to execute the solutions. The coding was done in *java* and a student licensed *IBM CPLEX* external library file is used to execute the ILP. 8 different bus systems available from MatPower with number buses $24, 30, 39, 57, 89, 118, 145, 300$ were used to generate the dependency equations for power network (using the rules described in Section 3.1.1). The time to generate the dependency equations were less than $2ms$. Within the Maricopa county 4 disjoint regions were considered labeled as Region 1 through 4. Dependency equations for the interdependent power-communication

Algorithm 2: Heuristic Solution to the ENH Problem

```
Data: A system $\mathcal{I}(E, \mathcal{F}(E))$, set of entities $E'$ failing initially with $|E'| = K$ and
      hardening budget $k$.
Result: Set of hardened entities $\mathcal{H}$.
begin
    Initialize $\mathcal{H} \leftarrow \emptyset$ and $\mathcal{D} \leftarrow \emptyset$;
    Update $\mathcal{F}(E)$ as follows — (a) let $Q$ be the set of entities that does not fail on
      failing $K$ entities, (b) remove IDRs corresponding to entities in set $Q$, (c)
      update the minterm of remaining IDRs by removing entities in set $Q$;
    Update $E \leftarrow E \setminus Q$ ;
    while (|$\mathcal{H}$| is not equal to k) do
        For each entity $e_i \in E \backslash \mathcal{D}$ compute $PS(e_i|E')$ and $CFMHV(e_i)$;
        if There exists multiple entities having same value of highest cardinality of
          the set $PS(e_i|E')$ then
            Let $e_p$ be an entity having highest $CFMHV(e_p)$ among all $e_p$'s in the set
              of entities having highest cardinality of the set $PS(e_i|A' \cup B')$;
            If there is a tie choose arbitrarily;
            Update $\mathcal{H} \leftarrow \mathcal{H} \cup \{e_p\}$, $\mathcal{D} \leftarrow \mathcal{D} \cup PS(e_p|E')$;
            Update $\mathcal{F}(E)$ by removing entities in $PS(e_p|E')$ both in the left and
              right side of the IDRs ;
        else
            Let $e_i$ be an entity having highest cardinality of the set $PS(e_i|E')$;
            Update $\mathcal{H} \leftarrow \mathcal{H} \cup \{e_i\}$, $\mathcal{D} \leftarrow \mathcal{D} \cup PS(e_i|E')$;
            Update $\mathcal{F}(E)$ by removing entities in $PS(e_i|E')$ both in the left and right
              side of the IDRs ;
return $\mathcal{H}$ ;
```

network were generated for these regions using the rules described in Section 3.1.2. The number of entities in each of the 12 data sets are enumerated in Table 4.4. To determine the initially failing entities we used the ILP solution of $\mathcal{K}$ most vulnerable entities in [18]. The $\mathcal{K}$ most vulnerable entities problem finds a set of $\mathcal{K}$ entities in a system $\mathcal{I}(E, \mathcal{F}(E))$ which when failed at $t = 0$ causes the maximum number of entities to fail. For a given data set representing a system $\mathcal{I}(E, \mathcal{F}(E))$, the initially failing entities was taken as a set E' ($|E'| = K$) such that — (a) The set E' constitutes the K most vulnerable entities in the system, (b) Failing the entities in set E' would cause failure of at least $|E|/2$ entities in total. The cardinality of the set E' along with the total number of entities failed are enumerated in Table 4.4.

DataSet	**Num. Of Entities**	K	**Num. of Entities Killed**
24 bus	58	8	29
30 bus	71	13	36
39 bus	84	17	42
57 bus	135	26	68
89 bus	295	78	147
118 bus	297	89	149
145 bus	567	191	284
300 bus	709	145	355
Region 1	48	6	26
Region 2	46	8	23
Region 3	48	6	24
Region 4	53	8	27

Table 4.4: Number of Entities, K Value Chosen and Number of Entities Failed when the K Vulnerable Entities are Failed Initially for Different Data Sets

In comparing the ILP and heuristic solution of the ENH problem we considered 5 distinct hardening budgets for each data set. With K being the number of initially failing entities in a data set the hardening budgets were chosen between $[1, K-1]$ (with value of K obtained from Table 4.4). It is also ensured that the hardening budgets chosen had a high variance. Figures 4.1 - 4.12 shows the total number of entities protected from failure for each data set using the ILP and heuristic solution. The run-time performance of the solutions are provided in Table 4.5 (in the table 'Heu' refers to the heuristic solution and Hi refers to the hardening budget corresponding to the i^{th} budget from left used in the bar graph plots). From Figures 4.1 - 4.12 it can be seen that the heuristic performs almost similar to that of the ILP solution in terms of quality. The maximum percent difference of the total number of entities protected in the ILP when compared to the heuristic solution occurs for a hardening budget of 39 in the 145 bus system (Figure 4.7) with the percent difference being 3.1%. In terms of run-time, heuristic outperforms the ILP with the heuristic computing solutions nearly 200 times faster in larger systems (as seen for the 300 bus system in Table 4.5). Hence it can be reasonably argued that the heuristic produces fast near optimal solutions for the ENH problem.

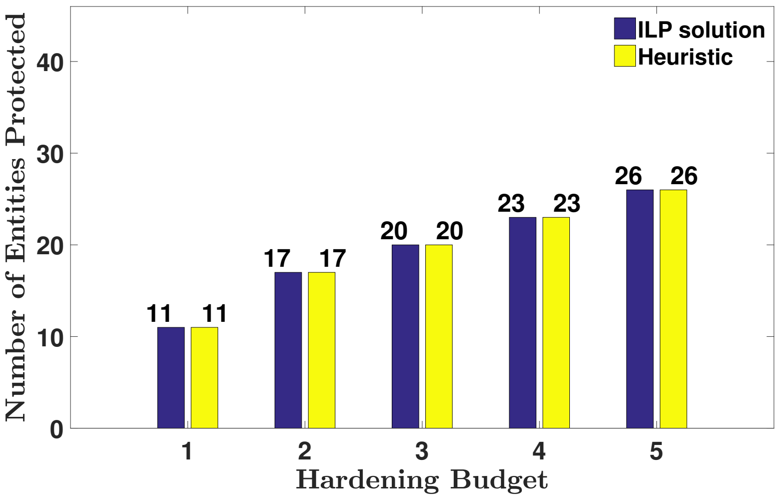

Figure 4.1: Comparison of ILP Solution with Heuristic for 24 Bus System (ENH Problem)

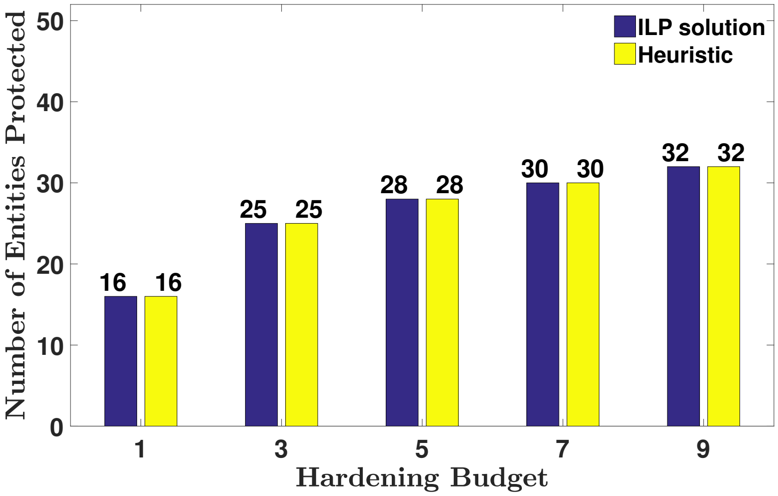

Figure 4.2: Comparison of ILP Solution with Heuristic for 30 Bus System (ENH Problem)

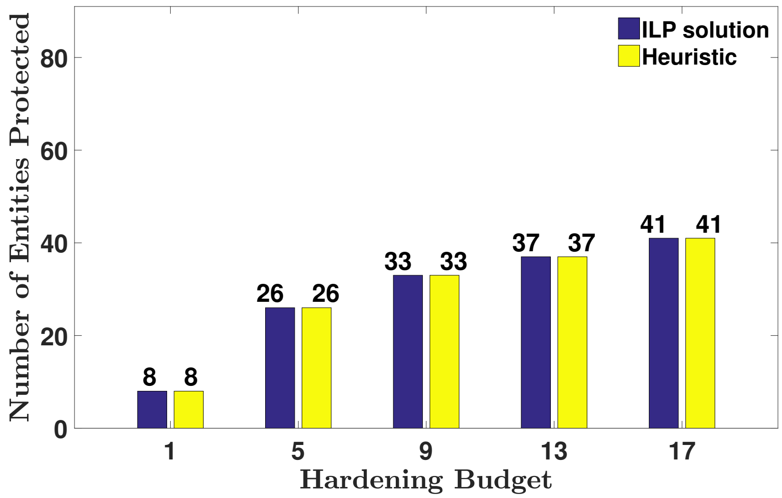

Figure 4.3: Comparison of ILP Solution with Heuristic for 39 Bus System (ENH Problem)

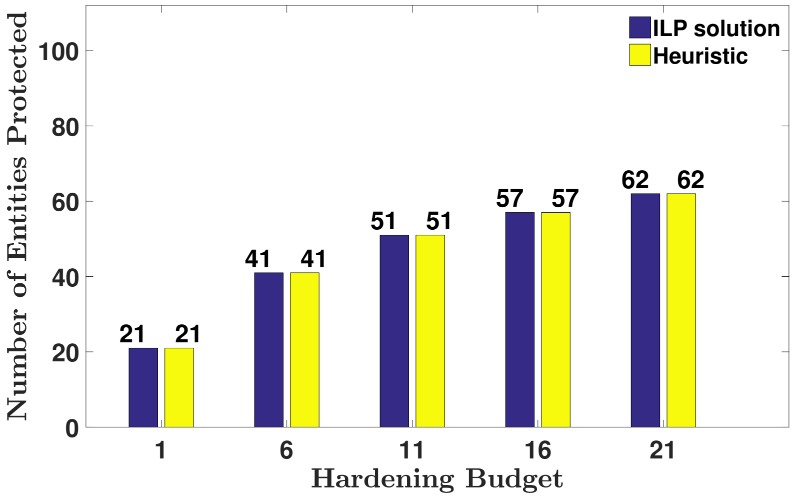

Figure 4.4: Comparison of ILP Solution with Heuristic for 57 Bus System (ENH Problem)

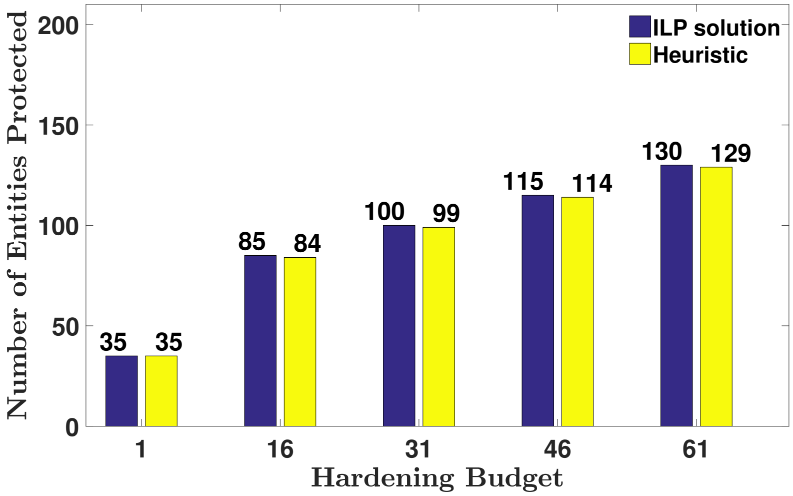

Figure 4.5: Comparison of ILP Solution with Heuristic for 89 Bus System (ENH Problem)

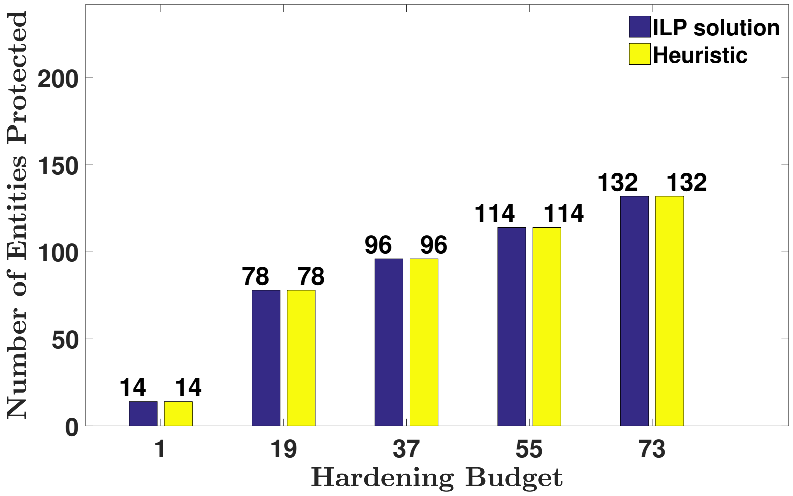

Figure 4.6: Comparison of ILP Solution with Heuristic for 118 Bus System (ENH Problem)

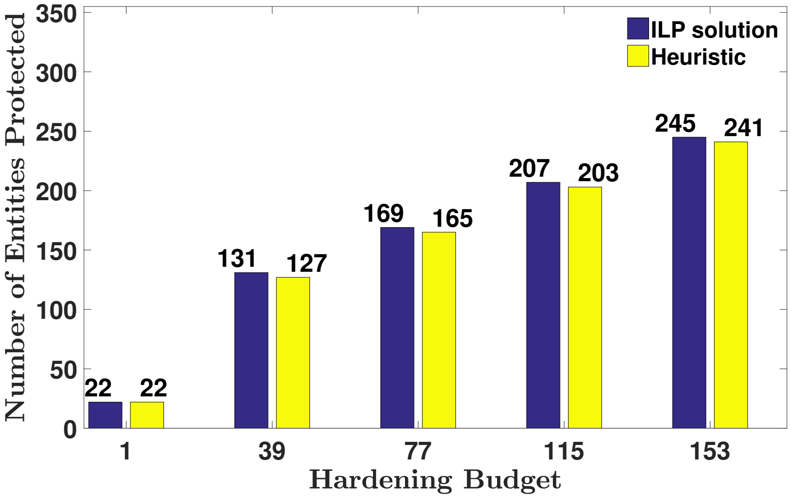

Figure 4.7: Comparison of ILP Solution with Heuristic for 145 Bus System (ENH Problem)

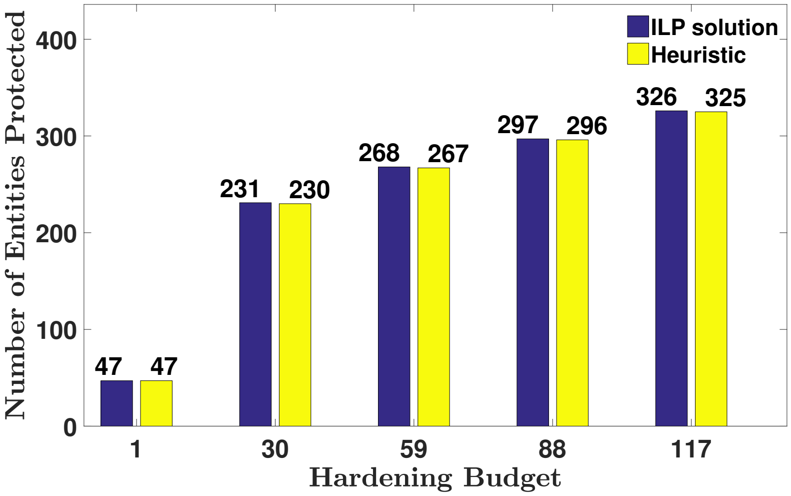

Figure 4.8: Comparison of ILP Solution with Heuristic for 300 Bus System (ENH Problem)

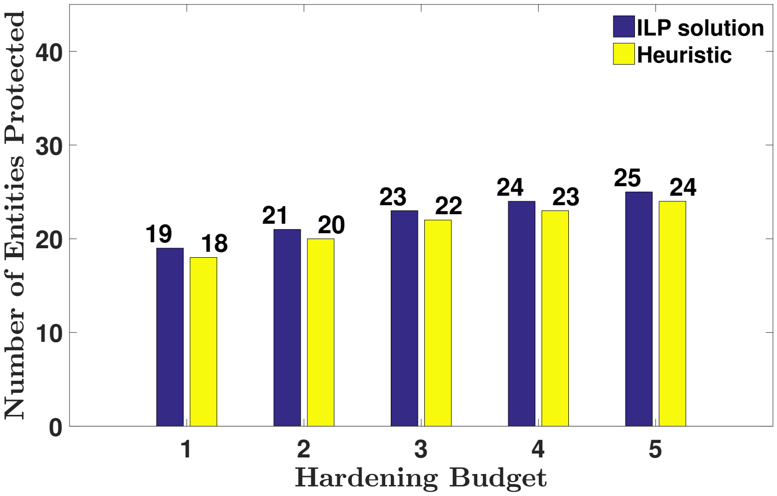

Figure 4.9: Comparison of ILP Solution with Heuristic for Region 1 (ENH Problem)

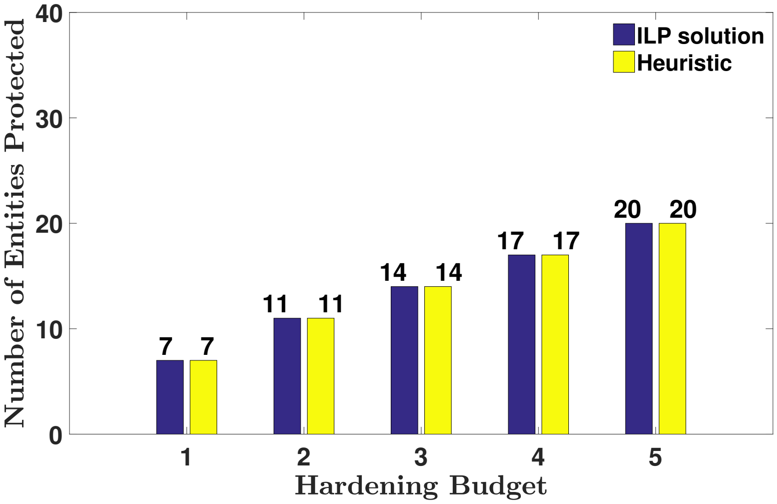

Figure 4.10: Comparison of ILP Solution with Heuristic for Region 2 (ENH Problem)

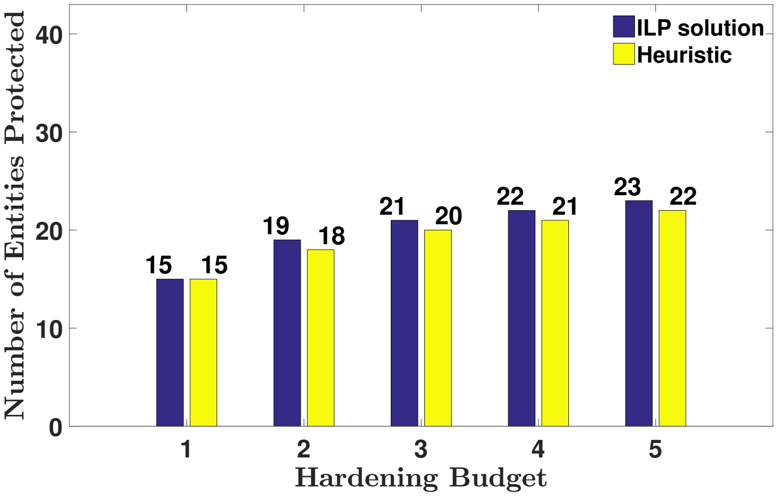

Figure 4.11: Comparison of ILP Solution with Heuristic for Region 3 (ENH Problem)

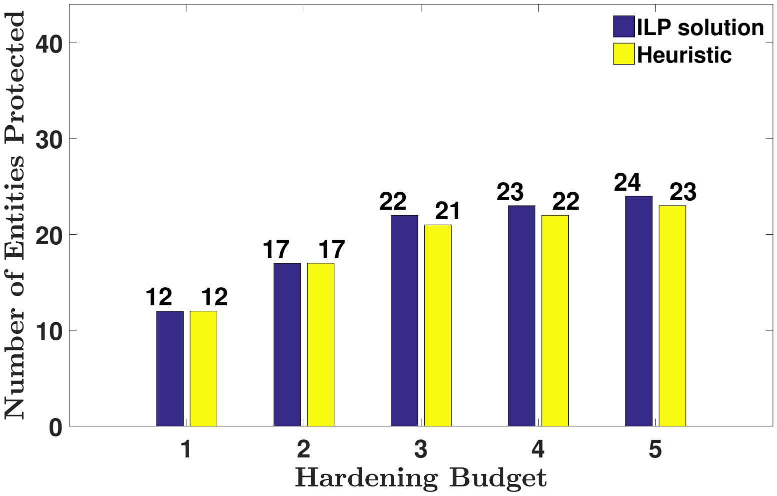

Figure 4.12: Comparison of ILP Solution with Heuristic for Region 4 (ENH Problem)

DataSet	Running time (in sec)									
	H1		H2		H3		H4		H5	
	ILP	Heu	ILP	Heu	ILP	Heu	ILP	Heu	ILP	Heu
24 bus	0.45	0.01	0.25	0.01	0.72	0.01	0.23	0.01	0.21	0.01
30 bus	2.44	0.01	0.38	0.01	0.38	0.01	0.35	0.01	0.34	0.01
39 bus	0.80	0.01	0.50	0.01	0.49	0.01	0.48	0.01	0.47	0.01
57 bus	2.67	0.03	1.73	0.01	2.21	0.01	2.27	0.01	1.68	0.01
89 bus	23.2	0.05	14.6	0.03	14.6	0.03	14.5	0.03	14.7	0.75
118 bus	20.9	0.04	16.2	0.06	17.2	0.09	17.1	0.02	17.1	0.02
145 bus	85.2	0.05	71.0	0.10	71.3	0.18	68.4	0.06	78.3	0.07
300 bus	282	0.15	222	1.56	217	0.85	253	0.39	264	0.40
Region 1	0.53	0.01	0.36	0.01	0.34	0.01	0.36	0.01	0.36	0.01
Region 2	13.8	0.01	12.9	0.01	12.8	0.01	13.1	0.01	13.2	0.01
Region 3	1.92	0.01	1.36	0.01	1.29	0.01	1.31	0.01	1.44	0.01
Region 4	1.48	0.01	1.43	0.01	1.10	0.01	1.06	0.01	1.05	0.01

Table 4.5: Run Time Comparison of Integer Linear Program and Heuristic for Different Data Sets (ENH Problem)

Chapter 5

THE TARGETED ENTITY HARDENING PROBLEM

The *Targeted Entity Hardening* is a restricted version of the *Entity Hardening Problem*. For an intra-dependent power network or interdependent power and communication network, certain entities might have higher priority to be protected. There might exist entities whose non-functionality poses higher economic or societal damage as compared to other entities. For example, power and communication network entities corresponding to office buildings running global stock exchanges, the U.S. White House, transportation sectors like airports etc. presumably are more important to be protected. Let F denote the failed set of entities (including initial and induced failure) when a set of K entities fail initially. We define a set P (with $P \subseteq F$) of entities which have a higher priority to be protected. The *Targeted Entity Hardening* problem finds the minimum set of entities which when hardened would ensure that none of the entities in set P fail.

5.1 Problem Formulation

Qualitatively, for a system $\mathcal{I}(E, \mathcal{F}(E))$ the objective of the Targeted Entity Hardening problem is to choose a minimum cardinality set of entities to harden, with a set of initially failing entities, such that all entities in a given set P are protected from failure. We use the example with dependency equations outlined in Table 3.1 of Chapter 3 to describe the Targeted Entity Hardening Problem with $P = \{b_4\}$. With $\{a_2, a_3\}$ being the two entities failing initially, hardening entity a_2 (with a_3 failing) would prevent failure of entities a_1, a_3, b_1, b_1, b_3. Similarly, hardening the entity a_3 (with a_2 failing) would prevent the failure of entity b_4. Even though hardening a_2

prevent failure of more entities than hardening a_3, owing to the problem description a_3 has to be hardened which is a solution to the Targeted Entity Hardening problem in this scenario. It is to be noted that other entities might also be protected from failure when a set of entities are hardened to protect a given set of entities. The Targeted Entity Hardening problem is formally stated below accompanied with a descriptive diagram provided in Figure 5.1 (in the figure direct failure means initial failure) —

The Targeted Entity Hardening (TEH) problem

INSTANCE: Given:

(i) A system $\mathcal{I}(E, \mathcal{F}(E))$, where the set E represent the set of entities, and $\mathcal{F}(E)$ is the set of IDRs.

(ii) The set of K entities failing initially E', where $E' \subseteq E$.

(iii) The set $F \subseteq E$ contains all the entities failed due to initial failure of E' entities i.e. $KillSet(E')$

(iv) A positive integer k and $k < K$.

(v) A set $P \subseteq F$.

DECISION VERSION: Is there a set of entities $H = E'' \subseteq E, |H| \leq k$, such that hardening H entities would result in protecting all entities in the set P after entities in E' fail at the initial time step.

OPTIMIZATION VERSION: Find the minimum set of entities in E to harden that would result in protecting all entities in the set P after entities in E' fail at the initial time step.

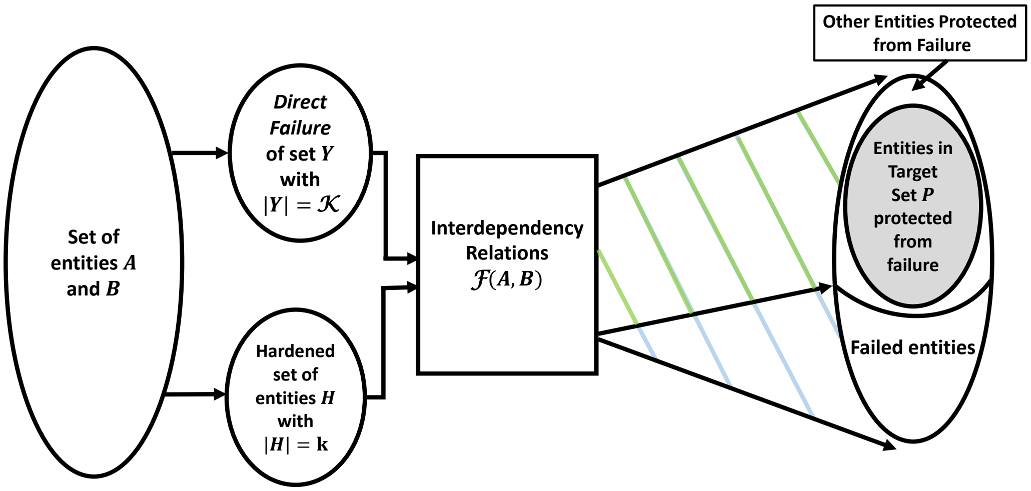
Set of entities A and B
$\boldsymbol{Direct\ Failure}$ of set Y with $|Y| = \mathcal{K}$
Hardened set of entities H with $|H| = \mathbf{k}$
Interdependency Relations $\mathcal{F}(A, B)$
Other Entities Protected from Failure
Entities in Target Set P protected from failure
Failed entities

Figure 5.1: Pictographic Description of the Targeted Entity Hardening Problem

5.2 Computational Complexity Analysis

In this subsection we prove the computational complexity of the Targeted Entity Hardening Problem to be NP-complete in Theorem 7.

Theorem 7. *The TEH problem is NP-complete*

Proof. We proof that the Targeted Entity Hardening is NP complete by a reduction from Set Cover problem. An instance of the Set Cover problem consists of (i) a set of elements $U = \{x_1, x_2, \ldots, x_n\}$, (ii) a set of subsets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ with $S_i \subseteq U$ $\forall S_i \in \mathcal{S}$, and (iii) a positive integer M. The problem asks the question whether there is a subset $\mathcal{S}'$ of $\mathcal{S}$ with $|\mathcal{S}'| \leq M$ such that $\bigcup_{S_k \in \mathcal{S}'} S_k = U$. From an instance of the Set Cover problem we create an instance of the Targeted Entity Hardening Problem as follows. Consider a system $\mathcal{I}(E, \mathcal{F}(E))$ with $E = A \cup B$, where A and B are entities of two separate critical infrastructures dependent on each other. For each element x_j in U we add an entity a_j in set A. Similarly for each subset S_i in set $\mathcal{S}$ we add an entity b_i in set B. For each element $x_i \in U$ which appears in subsets $S_m, S_n, S_p \in \mathcal{S}$

(say) we add an IDR $a_i \leftarrow b_m + b_n + b_p$ to $\mathcal{F}(E)$. There are no IDRs for entities in set B which prevents any cascading failure. The value of K is set to $|\mathcal{S}|$ and $E' = B$ which fails all entities in $A \cup B$. The set of P entities to be protected is set to A and k is set to M.

Consider there exists a solution to the Set Cover problem. Then there exist a set $\mathcal{S}'$ of cardinality M such that $\bigcup_{S_k \in \mathcal{S}'} S_k = U$. For each subsets $S_k \in \mathcal{S}'$ we harden the entity $b_k \in B$. So in each IDR of the A type entities there exist a B type entity that is hardened. Hence all A type entities will be protected from failure thus solving the Targeted Entity Hardening problem.

On the other way round consider there is a solution to the Targeted Entity Hardening problem. This ensures either that for each entity $a_j \in A$ (i) a_j itself is hardened, or (ii) at least one entity from set B in a_j's IDR is hardened. For scenario (i) arbitrarily select an entity b_p in a_j's IDR and include it in set C. For scenario (ii) include the hardened entities in the IDR of a_j into set C. This is done for each entity $a_j \in A$. For each entity in set C select the corresponding subset in set $\mathcal{S}$. The union of these set of subsets would result in the set U. Thus solving the set cover problem. Hence the theorem is proved. □

5.2.1 Restricted Case I: Problem Instance with One Minterm of Size One

This restriction imposed on the IDRs is the same as that of restricted case I of the ENH problem. Using the definition of *Protection set* (as in Section 4.2.2) and the result in Theorem 8 we design an algorithm (Algorithm 3) that solves the problem for this restricted case optimally in polynomial time (proved in Theorem 9).

Theorem 8. *Given a system $\mathcal{I}(E, \mathcal{F}(E))$ with IDRs of form restricted case I and $E' \subset E$ entities failing initially, for any entity e_i and e_j with $e_i \neq e_j$ either (a) $PS(e_i|E') \subseteq PS(e_j|E')$, (b) $PS(e_j|E') \subseteq PS(e_i|E')$, or (c) $PS(e_i|E') \cap PS(e_j|E') =$*

$\emptyset$.

Proof. Consider a directed graph $G = (V, E_D)$. The vertex set V consists of a vertex for each entity in E. For each IDR of form $y \leftarrow x$ there is a directed edge $(x, y) \in E_D$. In this proof the term vertex and entity is used interchangeably as an entity is essentially a vertex in G. It can be shown that G is either (a) Directed Acyclic Graph (DAG) with maximum in-degree of at most 1 or, (b) contain at most one cycle with no incoming edge to any vertex in the cycle and maximum in-degree of at most 1, or (c) collection of graphs (a) and/or (b). Consider a vertex $x_i \in V$. Let $G' = (V', E'_D)$ be a subgraph of G with V' consisting of x_i and all the vertices that has a directed path from x_i. Moreover, the edge set E'_D consists of all edges $(x, y) \in E_D$ with $x, y \in V'$ except for any edge (y, x_i) with $y_i \in V'$. Such a subgraph G' would be a directed tree with (i) one or more entities in $V' \backslash \{x_i\}$ is in $A' \cup B'$. Let X denote the set of such entities which satisfy this property, or (ii) no entities in $V' \backslash \{x_i\}$ is in E'. If the entity x_i is hardened then for case (i) all the entities in V' would be protected from failure except for entities in all subtrees with roots in X. The set of entities in such subtrees are contained in a set Z (say). For this condition if $e_j \in V' \backslash Z$ then $PS(e_j|E') \subset PS(e_i|E')$. Else if $e_j \in Z$ then $PS(x_i|E') \cap PS(e_j|E') = \emptyset$. For case (ii) for any entity $x_j \in V'$ the condition $PS(e_j|E') \subseteq PS(E_i|E')$ always holds (the equality holds for graphs of type (b) as stated above). This property holds for all entities in the entity set E. Hence proved. □

Theorem 9. *Algorithm 3 solves the Targeted Entity Hardening problem with IDRs having single minterms of size* 1 *optimally in polynomial time.*

Proof. The *Protection Sets* of the entities can be found in a similar way as that of computing *Kill Sets* defined in [18]. It can be shown that computing these sets for all entities in E can be done in $\mathcal{O}(n^3)$ where $n = |E|$. The while loop

Algorithm 3: Algorithm for TEH problem with IDRs in form of Restricted Case I

```
Data: A system $\mathcal{I}(E, \mathcal{F}(E))$, set $E'$ with $|E'| = K$ entities failing initially and the
      set $P$ of entities to be protected from failure.
Result: A set of entities $H$ to be hardened.
1  begin
2      For each entity $e_i \in (E)$ compute the Protection Sets $PS(e_i|E')$ ;
3      Initialize $H = \emptyset$ ;
4      while $P \neq \emptyset$ do
5          Choose the Protection Set with highest $|PS(e_i|E') \cap P|$;
6          Update $H \leftarrow H \cup \{e_i\}$ ;
7          Update $P \leftarrow P \backslash PS(e_i|E')$;
8          for all $d_j \in E$ do
9              $PS(e_j|E') = PS(e_j|E') \backslash PS(e_i|E')$;
10     return $H$ ;
```

in Algorithm 3 iterates for a maximum of n times. Step 5 can be computed in $\mathcal{O}(n^2)$ time. The for loop in step 8 iterates for n times. For any given e_j and e_i, $PS(e_j|E') = PS(e_j|E') \backslash PS(e_i|E')$ can be computed in $\mathcal{O}(n^2)$ time with the worst case being the condition when $|PS(e_i|E')| = |PS(e_j|E')| = n$. As step 9 is nested in a for loop within the while loop this accounts for the most expensive step in the algorithm. The time complexity of this step is $\mathcal{O}(n^4)$. Thus Algorithm 3 runs polynomially in n with time complexity being $\mathcal{O}(n^4)$.

In Algorithm 3 the while loop iterates till all the entities in P are protected from failure. In step 5 the entity e_i with protection set $PS(e_i|E')$ having most number of entities belonging to set P is chosen to be hardened. Correspondingly the entity e_i

is added to the hardening set H. The set P is updated by removing the entities in $PS(e_i|E')$. Similarly all the protection sets are updated by removing the entities in $PS(e_i|E')$.

We use the result from Theorem 8 to prove the optimality of Algorithm 3. An entity e_i is selected to be hardened at any iteration of the while loop has maximum number of entities in $PS(e_i|E') \cap P$. All entities e_j with $PS(e_j|E') \subseteq PS(e_i|E')$ would have $PS(e_j|E') \cap P \subseteq PS(e_i|E') \cap P$. Moreover there exist no entity e_k for which $PS(e_i|E') \subset PS(e_k|E')$ otherwise e_k would have been hardened instead. Hence there exist no other entity that protect other entities in P including $PS(e_i|E') \cap P$. So Algorithm 3 selects the minimum number of entities to harden that protects all entities in P. □

5.2.2 Restricted Case II: IDRs Having Arbitrary Number of Minterms of Size 1

For instance created in Theorem 7 the IDRs were logical disjunctions of minterms with size 1. We consider this restriction to design an approximation algorithm for the TEH problem and is shown in Theorem 10.

Theorem 10. *The Targeted Entity Hardening Problem is $\mathcal{O}(log(|P|)$ approximate when IDRs are logical disjunctions of minterms with size* 1.

Proof. We first compute the protection set $PS(e_i|E')$ for all entities $e_i \in E$. Each protection set is pruned by removing entities that are not in set P. Now the Targeted Entity Hardening Problem can be directly transformed into Minimum Set Cover problem by setting $U = P$ and $\mathcal{S} = \{PS(e_1|E'), PS(e_2|E'), ..., PS(x_|E||E')\}$. Selecting the corresponding entities of the protection sets that solve the Minimum Set Cover problem would also solve the Targeted Entity Hardening problem. There exists an approximation ratio of order $\mathcal{O}(log(n))$ (where n is the number of elements in set U) for the Set Cover problem. Therefore using the approximation algorithm that solves the Set Cover problem, the same ratio holds for the Targeted Entity Hardening problem with $n = |P|$. Hence proved. □

5.3 Optimal and Heuristic Solution to the Problem

In this section an optimal ILP solution and a sub-optimal with polynomial time complexity heuristic solution are described for the TEH problem.

5.3.1 Optimal solution using Integer Linear Program

The ILP formulation of the TEH problem is similar to that of ENH problem. The only difference being there is no hardening budget in TEH problem and additionally there is a set $P \subset E$ of entities that should be protected from failure. We use the same notations as of the ILP that solves the ENH problem. Using this the objective of the TEH problem is formulated as follows:

$$min\Big(\sum_{i=1}^{n} q_{x_i}\Big) \tag{5.1}$$

The constraint sets 2,3, and 4 of the ENH problem is employed in the TEH problem as well along with an additional constraint set as described below:

Additional Constraint Set: For all entities $e_i, \in P$, $x_{i(n-1)} = 0$. This ensures that all the entities in set P are protected from failure at the final time step.

With these constraints, the objective in (5.1) minimizes the number of hardened entities that results in protection of all entities in set P.

5.3.2 Heuristic Solution

In this subsection we provide a greedy heuristic solution to the TEH problem. For a given system $\mathcal{I}(E, \mathcal{F}(E))$ with set of entities as $E'(|E'| = K)$ failing initially and set of entities to protet being P, a heuristic is developed based on the following two metrics — (a) *Protection Set* as defined in Section 4.2.2, (b) *Prioritized Cumulative Fractional Minterm Hit Value (PCFMHV).*

Definition: The Prioritized Fractional Minterm Hit Value of an entity $e_j \in E$ in a system $\mathcal{I}(E, \mathcal{F}(E))$ is denoted as $FMHV(e_j, X)$. It is calculated as $PFMHV(e_j, P) = \sum_{i=1}^{m} \frac{1}{|s_i|}$. In the formulation m are the minterms in which e_j appears over IDRs in non operational entities in set P. The parameter s_i denotes i^{th} such minterm. If the e_j is hardened (or protected from failure) the value computed provides an estimate future impact on protection of other non operational entities in set P.

Definition: The Prioritized Cumulative Fractional Minterm Hit Value of an entity $e_j \in E$ is denoted as $PCFMHV(e_j)$. It is computed as $PCFMHV(e_j) = \sum_{\forall x_i \in PS(e_j|E')} PFMHV(x_i, PS(x_i|E'))$. This gives a measure of future impact on protecting non functional entities in P when the entity e_j is hardened and entities $PS(e_j|E')$ are protected from failure.

Using these definitions, the heuristic is formulated in Algorithm 4. For each iteration of the while loop in the algorithm, the entity having highest cardinality of the set $PS(x_i|A' \cup B') \cap P$ is hardened. This ensures that at each step the number of entities protected in set P is maximized. In case of a tie, the entity having highest Prioritized Cumulative Fractional Minterm Hit Value among the set of tied entities is selected. This causes the selection of the entity that has the potential to protect maximum number of entities in updated set P in subsequent iterations. Thus, the

Algorithm 4: Heuristic solution to the TEH problem

```
Data: A system $\mathcal{I}(E, \mathcal{F}(E))$, set of $K$ vulnerable entities and the set $P$ of entities to
      be protected from failure.
Result: A set of entities $H$ to be hardened.
begin
    Initialize $\mathcal{D} = \emptyset$ and $H = \emptyset$ ;
    Update $\mathcal{F}(E)$ as follows — (a) let $Q$ be the set of entities that does not fail on
      failing $K$ entities, (b) remove IDRs corresponding to entities in set $Q$, (c)
      update the minterm of remaining IDRs by removing entities in set $Q$;
    while $P \neq \emptyset$ do
        For each entity $e_i \in E \backslash \mathcal{D}$ compute $PS(e_i|E')$ and $PCFMHV(e_i)$;
        if There exists multiple entities having same value of highest cardinality of
          the set $PS(e_i|E') \cap P$ then
            Let $e_p$ be an entity having highest $PCFMHV(e_p)$ among all $e_p$'s in the
              set of entities having highest cardinality of the set $PS(e_i|A' \cup B')$;
            If there is a tie choose arbitrarily;
            Update $H \leftarrow H \cup \{e_p\}$, $\mathcal{D} \leftarrow \mathcal{D} \cup PS(e_p|E')$, $P \leftarrow P \backslash PS(e_p|E')$;
            Update $\mathcal{F}(E)$ by removing entities in $PS(e_p|E')$ both in the left and
              right side of the IDRs ;
        else
            Let $e_i$ be an entity having highest cardinality of the set $PS(e_i|E') \cap P$;
            Update $H \leftarrow H \cup \{e_p\}$, $\mathcal{D} \leftarrow \mathcal{D} \cup PS(e_i|E')$, $P \leftarrow P \backslash PS(e_i|E')$;
            Update $\mathcal{F}(E)$ by removing entities in $PS(e_i|E')$ both in the left and right
              side of the IDRs ;
    return $H$ ;
```

heuristic greedily minimizes the set of entities hardened which would cause protection of all entities in P. The heuristic overestimates the cardinality of H from the optimal solution. Algorithm 4 runs in polynomial time, more specifically the time complexity is $\mathcal{O}(|P|n(n+m)^2)$ (where $n = |E|$ and $m =$ Number of minterms in $\mathcal{F}(E)$).

5.4 Experimental Result

To perform a comparative study of the heuristic with the ILP, we use the same data sets as outlined in Chapter 4. Additionally, the initial failure set is computed using $\mathcal{K}$ most vulnerable entities problem and the same value of $\mathcal{K}$ as in ENH problem are chosen for each data set (as in Table 4.4). 5 distinct protection sets P were considered for each data set. Let F denote the set entities failed in total when K entities fail initially. The cardinality of the set F and the value of K was taken from Table 4.4 for each data set. The cardinality of the protection set for a given data set was chosen between $[1, |F| - 1]$ ensuring that the chosen values have high variance. For a given cardinality C the protection set P was constructed by choosing C entities from the set F corresponding to a particular data set. Figures 5.2 - 5.13 shows the comparison of the Heuristic solution with the ILP in terms of total number of entities hardened for a given cardinality of protection budget. The run-time comparison of the solutions are provided in Table 5.1. A maximum percent difference of 25% (ILP compared with Heuristic) in the number of entities hardened can be seen in Region 2 for a $|P|$ value of 13 (Figure 5.11). However, for most of the cases the heuristic produces near optimal or optimal solution. The heuristic also compute the solutions nearly 200 times faster than the ILP for larger systems as seen in Table 5.1. Hence it can be claimed that the heuristic solution to the TEH problem produces near optimal solution at a much faster time compared to the ILP solution.

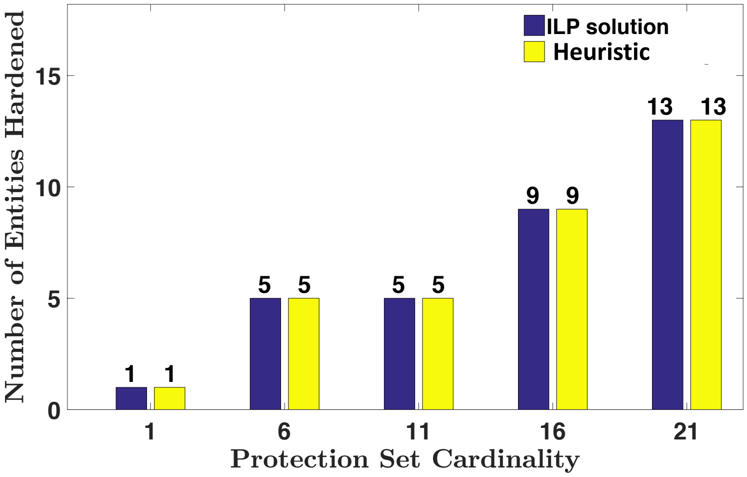

Figure 5.2: Comparison of ILP Solution with Heuristic for 24 Bus System (TEH Problem)

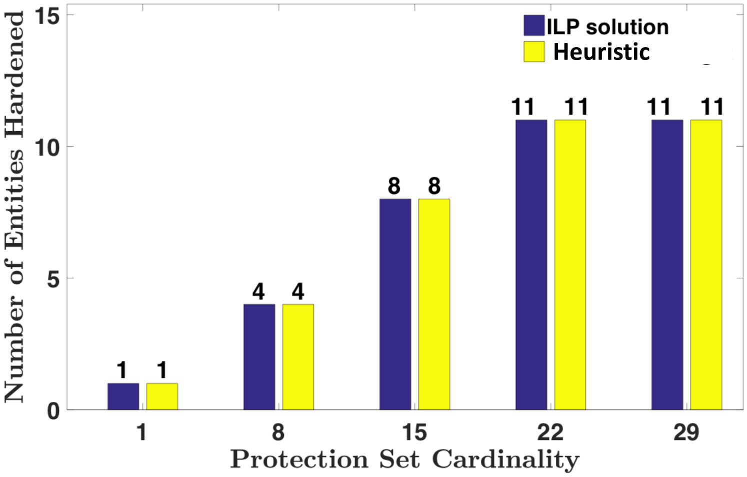

Figure 5.3: Comparison of ILP Solution with Heuristic for 30 Bus System (TEH Problem)

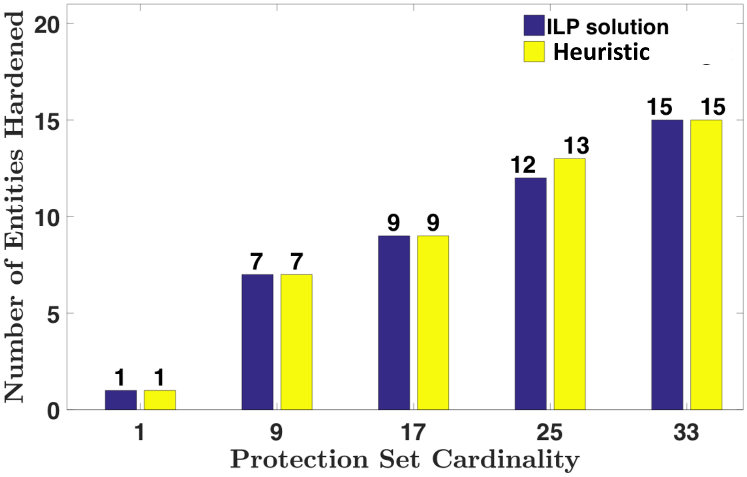

Figure 5.4: Comparison of ILP Solution with Heuristic for 39 Bus System (TEH Problem)

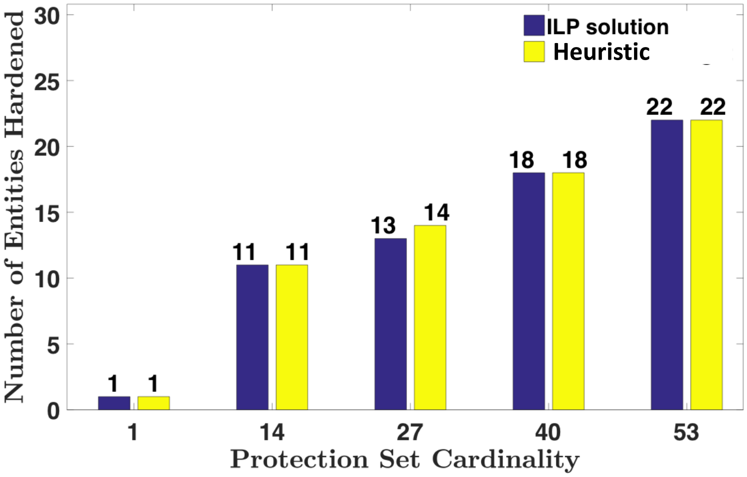

Figure 5.5: Comparison of ILP Solution with Heuristic for 57 Bus System (TEH Problem)

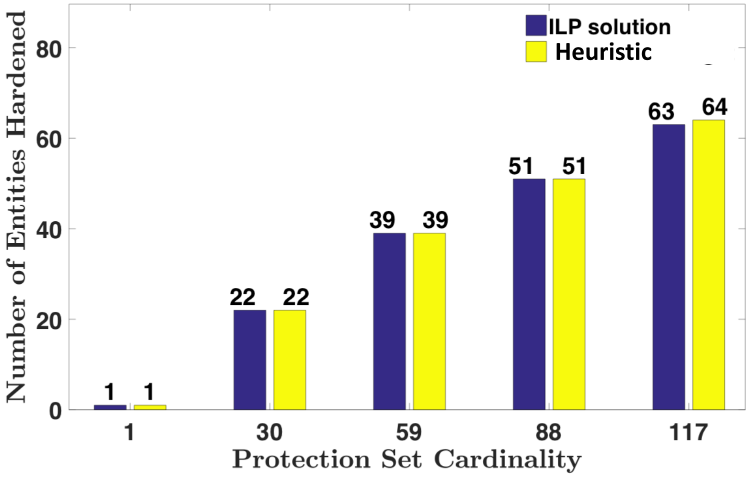

Figure 5.6: Comparison of ILP Solution with Heuristic for 89 Bus System (TEH Problem)

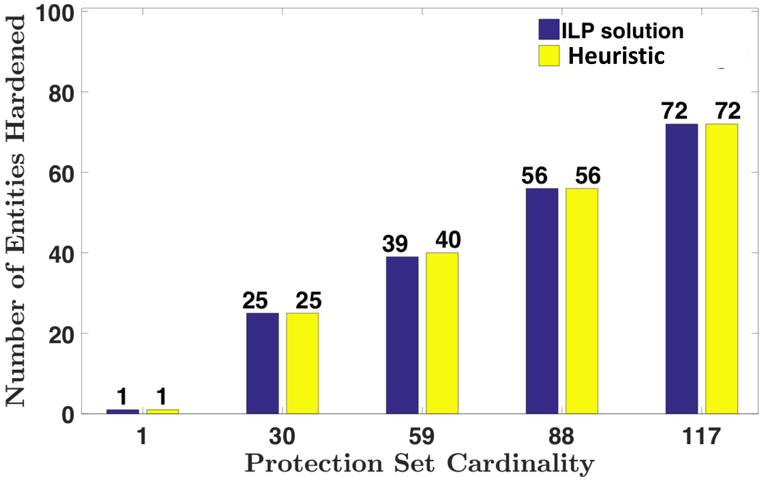

Figure 5.7: Comparison of ILP Solution with Heuristic for 118 Bus System (TEH Problem)

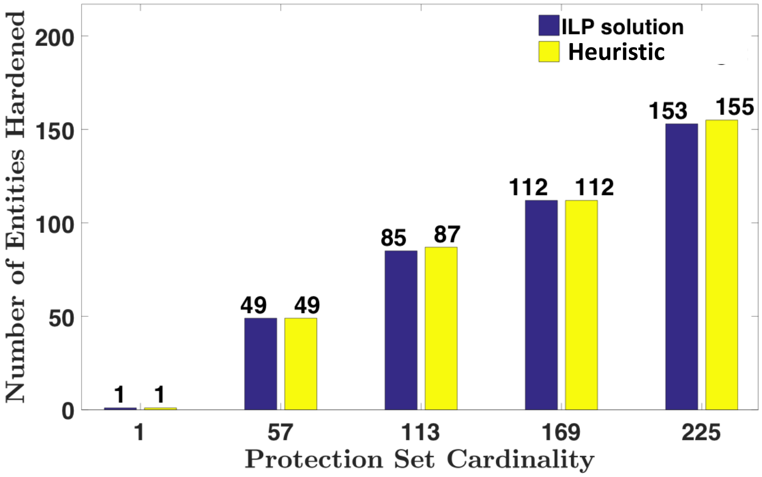

Figure 5.8: Comparison of ILP Solution with Heuristic for 145 Bus System (TEH Problem)

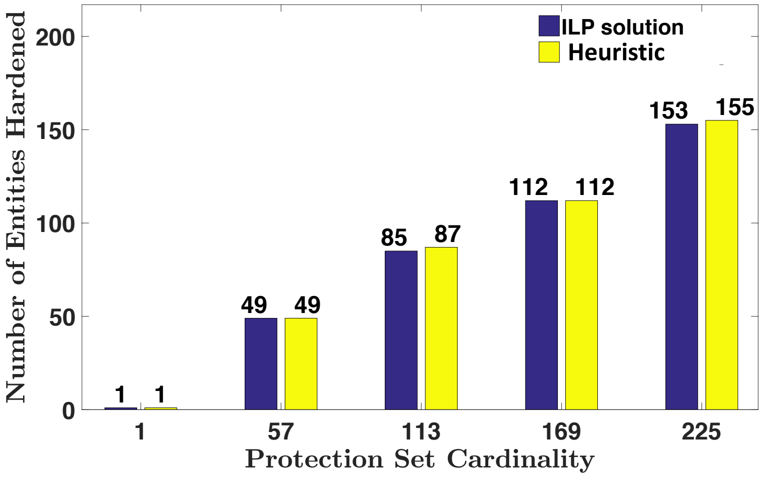

Figure 5.9: Comparison of ILP Solution with Heuristic for 300 Bus System (TEH Problem)

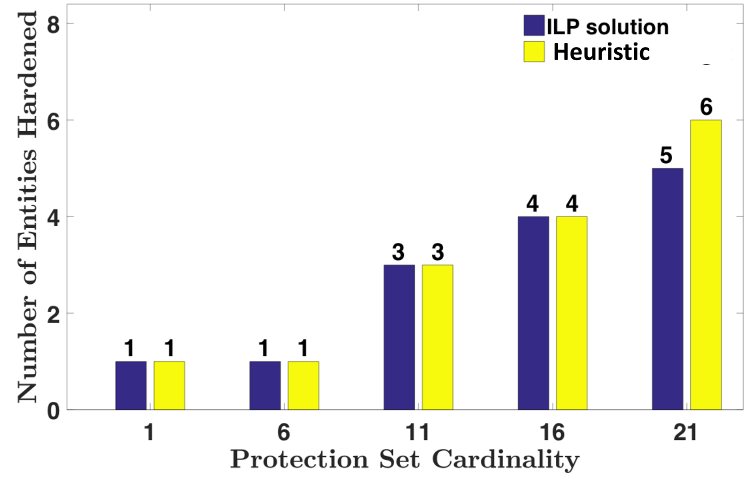

Figure 5.10: Comparison of ILP Solution with Heuristic for Region 1 (TEH Problem)

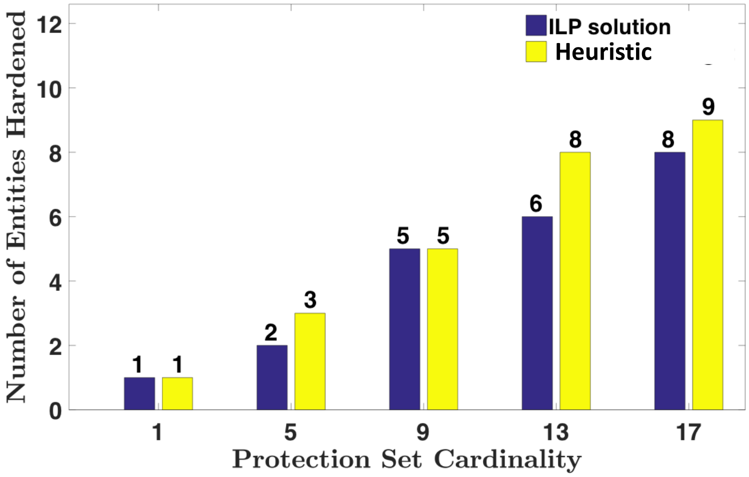

Figure 5.11: Comparison of ILP Solution with Heuristic for Region 2 (TEH Problem)

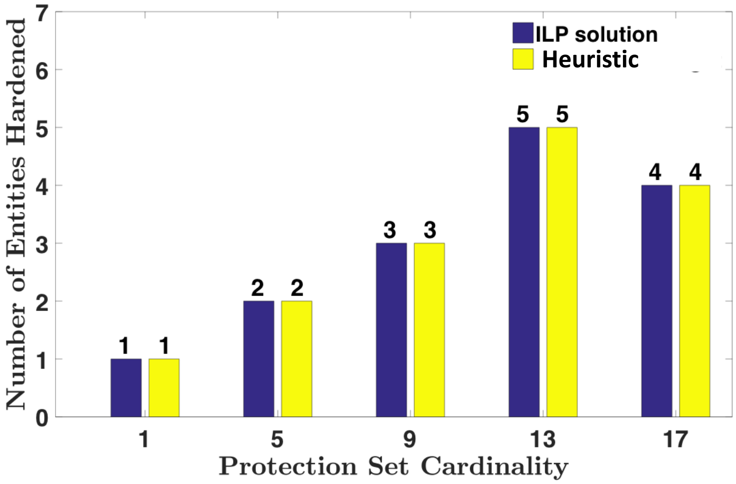

Figure 5.12: Comparison of ILP Solution with Heuristic for Region 3 (TEH Problem)

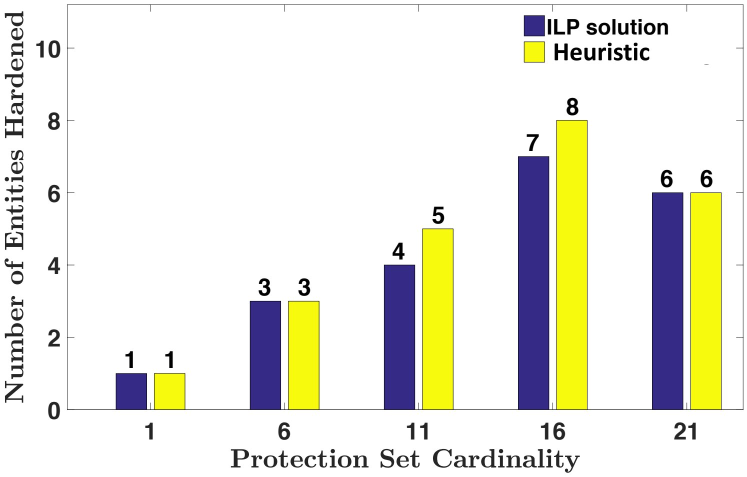

Figure 5.13: Comparison of ILP Solution with Heuristic for Region 4 (TEH Problem)

DataSet	**Running time (in sec)**									
	P1		**P2**		**P3**		**P4**		**P5**	
	ILP	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**
24 bus	0.42	0.01	0.22	0.01	0.20	0.01	0.19	0.01	0.19	0.01
30 bus	0.57	0.01	0.37	0.01	0.34	0.01	0.34	0.01	0.31	0.01
39 bus	0.82	0.01	0.49	0.01	0.49	0.01	0.46	0.01	0.47	0.01
57 bus	2.23	0.02	1.69	0.02	2.00	0.02	2.07	0.02	1.91	0.01
89 bus	17.3	0.05	14.4	0.08	14.4	0.16	14.0	0.11	14.1	0.07
118 bus	17.6	0.13	17.3	0.15	16.9	0.10	16.3	0.08	17.0	0.07
145 bus	79.0	0.06	76.6	0.26	77.2	0.27	75.7	0.24	74.9	0.25
300 bus	302	0.18	241	1.01	234	1.41	229	1.03	230	1.20
Region 1	0.55	0.01	0.40	0.01	0.43	0.01	0.35	0.01	0.34	0.01
Region 2	14.5	0.01	13.5	0.01	13.4	0.01	13.4	0.01	13.3	0.01
Region 3	1.58	0.01	1.40	0.01	1.29	0.01	1.29	0.01	1.29	0.01
Region 4	1.36	0.01	1.12	0.01	1.21	0.01	1.09	0.01	1.03	0.01

Table 5.1: Run Time Comparison of Integer Linear Program and Heuristic for Different Data Sets (TEH Problem)

Chapter 6

THE AUXILIARY ENTITY ALLOCATION PROBLEM

For a given set of entities failing initially, the system reliability can be increased (i.e. entities can be protected from failure) by *Entity Hardening*. On scenarios where entity hardening is not possible it is imperative to take alternative strategies. The number of entities failing due to *induced failure* can be reduced by modifying the IDRs. One way of modifying an IDR is adding an entity as a new minterm. For example, consider the system $\mathcal{I}(E, \mathcal{F}(E))$ with IDRs given by Table 6.1. The cascade propagation is hown in Table 6.2 when entities b_2 and b_3 fail initially. Let the IDR $b_1 \leftarrow a_2$ be modified as $b_1 \leftarrow a_2 + a_5$. Hence the new system is represented as $\mathcal{I}(E, \mathcal{F}'(E))$ with the same set of IDRs as that in Table 6.1 except for IDR $b_1 \leftarrow a_2 + a_5$ as the sole modification. The entity a_1 introduced is termed as an *auxiliary entity.* It follows that after the modification, failure of entities b_2 and b_3 at time $t = 0$ would trigger failure of entities a_2, a_3 and a_4 only. Thus before modification the failure set would have been $\{a_1, a_2, a_3, a_4, b_1, b_2, b_3\}$ and after the modification it would be $\{a_2, a_3, a_4, b_2, b_3\}$. Thus the modification would lead to a fewer number of failures.

We make the following assumptions while modifying an IDR —

- It is possible to add an auxiliary entity as conjunction to a minterm. However it is intuitive that this would have no impact in decreasing the number of entities failed due to *induced failure*. Hence we modify an IDR by adding only one auxiliary entity as a disjunction to a minterm

- An auxiliary entity does not have the capacity to make an entity operational which fails due to *initial failure*. So to prune the search set for obtaining a

Power Network	**Comm. Network**
$a_1 \leftarrow b_1 + b_2$	$b_1 \leftarrow a_2$
$a_2 \leftarrow b_1 b_2$	$b_2 \leftarrow a_2$
$a_3 \leftarrow b_2 + b_1 b_3$	$b_3 \leftarrow a_4$
$a_4 \leftarrow b_3$	$--$
a_5	$--$

Table 6.1: IDRs for the Constructed Example

Entities	Time Steps (t)							
	0	1	2	3	4	5	6	7
a_1	0	0	0	1	1	1	1	1
a_2	0	1	1	1	1	1	1	1
a_3	0	1	1	1	1	1	1	1
a_4	0	1	1	1	1	1	1	1
a_5	0	0	0	0	0	0	0	0
b_1	0	0	1	1	1	1	1	1
b_2	1	1	1	1	1	1	1	1
b_3	1	1	1	1	1	1	1	1

Table 6.2: Cascade Propagation when Entities $\{b_2, b_3\}$ Fail Initially. 0 Denotes the Entity is Operational and 1 Non-Operational

solution we discard entities in the set of initially failing entities as possible auxiliary entities.

- If an IDR D is modified then it is done by adding only one entity not in $E' \cup E_D$ where E' is the set of initially failing entities and E_D is a set consisting of all entities (both on left and right side of the equation) in D. For any IDR $D \in \mathcal{F}(E$ we denote $AUX = E/(E' \cup E_D)$ as the set of auxiliary entities that can be used to modify D.

With these definitions the Auxiliary Entity Allocation Problem (AEAP) is defined as follows. Let for a system $\mathcal{I}(E, \mathcal{F}(E))$, $E' \subset E$ be the set of initially failing entities. With a budget S in number of modifications, the task is to find which are the S IDRs that are to be modified and which entity should be used to perform this modification such that number of entities failing due to *induced failure* is minimized. A more formal description given below.

The Auxiliary Entity Allocation Problem (AEAP)

Instance — A system $\mathcal{I}(E, \mathcal{F}(E))$, set of entities $E' \subset E$ failing initially and two positive integers S and P_f.

Decision Version — Does there exist S IDR auxiliary entity tuple (D, x_i) such that when each IDRs $D \in \mathcal{F}(E)$ is modified by adding auxiliary entity $x_i \in AUX$ as a disjunction it would protect at least P_f entities from *induced failure* with entities in set E' failing initially.

6.1 Computational Complexity Analysis

In this section we analyze the computational complexity AEAP. The computational complexity of the problem depends on nature of the IDRs. The problem is first solved by restricting the IDRs to have one minterm of size 1. For this special case a polynomial time algorithm exists for the problem. With IDRs in general form the problem is proved to be NP-complete.

6.1.1 Special Case: Problem Instance with One Minterm of Size One

The special case consist of IDRs which have a single minterm of size 1 and each entity appearing exactly once on the right hand side of the IDR. AEAP can be solved in polynomial time for this case. We first define *Auxiliary Entity Protection Set* and use it to provide a polynomial time heuristic in Algorithm 5.

Definition: Auxiliary Entity Protection Set: With a given set of E' entities failing initially the Auxiliary Entity Protection Set is defined as the number of entities protected from *induced failure* when an auxiliary entity x_i is added as a disjunction to an IDR $D \in \mathcal{F}(E)$. It is denoted as $AP(D, x_i|E')$.

6.1.2 General Case: Problem Instance with an Arbitrary Number of Minterms of Arbitrary Size

In Theorem 11 we prove that the decision version of AEAP for general case is NP complete.

Theorem 11. *The decision version of AEAP for Case IV is NP-complete.*

Proof. The hardness is proved by a reduction from Set Cover problem. An instance of a set cover problem consists of a universe $U = \{x_1, x_2, ..., x_n\}$ of elements and set of subsets $\mathcal{S} = \{S_1, S_2, ..., S_m\}$ where each element $S_i \in \mathcal{S}$ is a subset of U. Given an integer X the set cover problem finds whether there are $\leq X$ elements in $\mathcal{S}$ whose union is equal to U. From an instance of the set cover problem we create an instance

Algorithm 5: Algorithm solving AEAP for IDRs with minterms of size 1

```
Data: A system $\mathcal{I}(E, \mathcal{F}(E))$ and set of $E'$ entities failing initially
Result: A set $D_{sol}$ consisting of IDR auxiliary entity doubles $(D, x_i)$ (with
        $|D_{sol}| = S$ and $P_f$ (denoting the entities protected from induced failure)
begin
    For each IDR $D \in \mathcal{F}(E)$ and each entity $x_i \in AUX$ (where
      $AUX = E/(E' \cup E_D))$ compute the Auxiliary Entity Protection Set
      $AP(D, x_i|E')$ ;
    Initialize $D_{sol} = \emptyset$ and $P_f = \emptyset$;
    while $S \neq 0$ do
        Choose the Auxiliary Entity Protection Set with highest $AP(x_i, D|E')$. In
          case of tie break arbitrarily. Let $D_{cur}$ be the corresponding IDR and $x_{cur}$
          the auxiliary entity;
        Update $D_{sol} = D_{sol} \cup (D_{cur}, x_{cur})$ and add auxiliary entity $x_{cur}$ as a
          disjunction to the IDR $D_{cur}$ ;
        Update $P_f = P_f \cup AP(D_{cur}, x_{cur}|E')$;
        for $\forall$ IDR $D' \in \mathcal{F}(E)$ and $x_i \in AUX$ of $D'$ do
            Update $AP(D', x_i|E') = AP(D', x_i|E') \backslash AP(D_{cur}, x_{cur}|E')$;
        $S \leftarrow S - 1$;
    return $D_{sol}$ and $P_f$ ;
```

of AEAP. Consider that the instance created refers to a system $\mathcal{I}(E, \mathcal{F}(E))$ where $E = A \cup B$ (A and B containing entities of two separate infrastructures). For each subset S_i we create an entity b_i and add it to set B. For each element x_j in U we add an entity a_j to a set A_1. We have a set A_2 of entities where $|A_2| = |B|$. Let $A_2 = \{a_{21}, a_{22}, ..., a_{2|B|}\}$ where there is an association between entity b_j and a_{2j}. Additionally we have a set of entities A_3 with $|A_3| = X$ which does not have any dependency relation of its own. The set A is comprised of $A_1 \cup A_2 \cup A_3$. The IDRs are created as follows. For an element x_i that appears in subsets S_x, S_y, S_z, an IDR $a_i \leftarrow b_x + b_y + b_z$ is created. For each entity $b_j \in B$ an IDR $b_j \leftarrow a_{2j}$ is added to $\mathcal{F}(E)$. The cardinality of E' is set to $|A_2|$ and it directly follows that $E' = A_2$. The value of S (number of IDR modifications) is set to X and P_f is set to $S + |A_1|$.

Let there exist a solution to the set cover problem. Then there exist at least X subsets whose union covers the set U. For each subset S_k which is in the solution of the set cover problem we choose the corresponding entity b_k. Let B' be all such entities. We arbitrarily choose and add an entity from A_3 to each IDR $b_k \leftarrow a_{2k}$ with $b_k \in B'$ to form $S = X$ distinct IDR auxiliary entity doubles. As A_3 type entities does not have any dependency relation thus all the entities in B that correspond to the subsets in the solution will be protected from failure. Additionally protecting these B type entities would ensure all entities in A_1 does not fail as well (as there exists at least one B type entity in the IDR of A_1 type entities which is operational). Hence a total of $X + |A_1|$ are protected from failure.

Similarly let there exist a solution to AEAP. It can be checked easily that no entities in $B \cup A_1 \cup A_2$ has the ability to protect additional entities using IDR modification. Hence set A_3 can only be used as auxiliary entities. An entity from A_3 for the created instance can be added to an IDR of A_1 type entity or B type entity. In the former strategy only one entity is protected from failure whereas two entities are operational when we add auxiliary entity to IDRs of B type entities. Hence all the auxiliary entities are added to the B type IDRs with a final protection of $X + |A_1|$ entities. For each IDR of the B type entity to which the auxiliary entity is added, the corresponding subset in $\mathcal{S}$ is chosen. The union of these subsets would result in U as the solution of AEAP that protects the failure of all A_1 type entities. Hence solving the set cover problem and proving the hardness stated in theorem. 11. □

6.2 Solutions to AEAP

We consider the following restricted case where there exists at least S entities in the system $\mathcal{I}(E, \mathcal{F}(E))$ which does not belong to any of the failing entities. This comprise the set of auxiliary entities that can be used. It is also imperative to use such set as auxiliary entities because they never fail from induced or initial failure when the entities in set E' fail initially. The problem still remains to be NP compete for this case as in Theorem 11 the set of entities A_3 belong to such class of auxiliary entities. With these definition of the special case let $\mathcal{A}$ denote a set of such auxiliary entities which can be used for IDR modifications with $\mathcal{A} \subset E/(E'')$ (where E'' are the entities that fails due to failing entities E' initially). Hence we loose the notion of IDR auxiliary entity doubles in the solution as any auxiliary entity from set $\mathcal{A}$ would produce the same protection effect. Let $\mathcal{A}$ denote all such entities that can be used as auxiliary entities as defined above. We additionally assume that $|\mathcal{A}| \geq S$, i.e., there are enough auxiliary entities to suffice the AEAP budget S. So in both the solutions we only consider the IDRs that needs to be modified and disregard which auxiliary entity is used for this modification. We first propose an Integer Linear Program (ILP) to obtain the optimal solution in this setting. We later provide a polynomial heuristic solution to the problem. The performance of heuristic with respect to the ILP is compared in the section to follow.

6.2.1 Optimal Solution to AEAP

We first define the variables used in formulating the ILP. A set of variables $G = \{g_1, g_2, ..., g_c\}$ (with $c = |E|$) is used to maintain the solution of E' most vulnerable entities. Any variable $g_i \in G$ is equal to 1 if $e_i \in E$ belongs to E' and is 0 otherwise. For each entity e_j a set of variables x_{jd} are introduced with $0 \leq d \leq |E| - 1$. x_{id} is set to 1 if the entity e_i is non operational at time step d and is 0 otherwise. Let P denote the total number of IDRs in the system and assume each IDR has a unique label between numbers from 1 to P. A set of variables $M = \{m_1, m_2, ..., m_P\}$ are introduced. The value of m_i is set to 1 if an auxiliary node is added as a disjunction to the IDR labeled i and 0 otherwise. With these definitions we define the objective function and the set of constraints in the ILP.

$$min\Big(\sum_{i=1}^{|E|} x_{i(|E|-1)}\Big) \tag{6.1}$$

The objective function defined in 6.1 tries to minimize the number of entities having value 1 at the end of the cascade i.e. time step $|E| - 1$. Explicitly this objective minimizes the number of entities failed due to induced failure. The constraints that are imposed on these objective to capture the definition of AEAP are listed below — *Constraint Set 1:* $x_{i0} \geq g_i$. This imposes the criteria that if entity e_i belongs to E' then the corresponding variable x_{i0} is set to 1 capturing the *initial failure*.

Constraint Set 2: $x_{id} \geq x_{i(d-1)}, \forall d, 1 \leq d \leq |E| - 1$. This ensures that the variable corresponding to an entity which fails at time step t would have value 1 for all $d \geq t$.

Constraint Set 3: We use the theory developed in [18] to generate constraints to represent the cascade through the set of IDRs. To describe this consider an IDR

$a_i \leftarrow b_j b_p b_l + b_m b_n + b_q$ in the system. Assuming the IDR is labeled v it is reformulated as $a_i \leftarrow b_j b_p b_l + b_m b_n + b_q + m_v$ with $m_v \in M$. This is done for all IDRs. The constraint formulation is described in the following steps.

Step 1: All minterms of size greater than 1 are replaced with a single virtual entity. In this example we introduce two virtual entities C_1 and C_2 ($C_1, C_2 \notin A \cup B$) capturing the IDRs $C_1 \leftarrow b_j b_p b_l$ and $C_2 \leftarrow b_m b_n$. The IDR in the example can be then transformed as $a_i \leftarrow C_1 + C_2 + b_q + m_v$. For any such virtual entity C_k a set of variables c_{kd} are added with $c_{kd} = 1$ if C_k is alive at time step d and 0 otherwise. Hence all the IDRs are represented as disjunction(s) of single entities. Similarly all virtual entities have IDRs which are conjunction of single entities.

Step 2: For a given virtual entity C_k and all entities having a single midterm of arbitrary size, we add constraints to capture the cascade propagation. Let N denote the number of entities in the IDR of C_k. The constraints added is described through the example stated above. The variable c_1 with IDR $C_1 \leftarrow b_j b_p b_l$, constraints $c_{1d} \geq \frac{y_{j(d-1)}+y_{p(d-1)}+y_{l(d-1)}}{N}$ and $c_{1d} \leq y_{j(d-1)} + y_{p(d-1)} + y_{l(d-1)} \forall d, 1 \leq d \leq |E| - 1$ are added (with $N = 3$ in this case). This ensures that if any entity in the conjunction fails the corresponding virtual entity fails as well.

Step 3: In the transformed IDRs described in step 1 let n denote the number of entities in disjunction for any given IDR (without modification). In the given example with IDR $a_i \leftarrow C_1 + C_2 + b_q + m_v$, constraints of form $x_{id} \geq c_{1(d-1)} + c_{2(d-1)} + y_{q(d-1)} + m_v - (n-1)$ and $x_{id} \leq \frac{c_{1(d-1)}+c_{2(d-1)}+y_{q(d-1)}+m_v}{n} \forall d, 1 \leq d \leq |E| - 1$ are added. This ensures that the entity a_i will fail only if all the entities in disjunction become non operational.

Constraint Set 4: To ensure that only S auxiliary entities are added as disjunction to the IDRs constraint $\sum_{v=1}^{P} m_v \leq S$ is introduced.

6.2.2 Heuristic solution to AEAP

In this section we provide a polynomial heuristic solution to AEAP. We first redenote *Auxiliary Entity Protection Set* as $AP(D|E')$ as it is immaterial which entity is added as an auxiliary entity since no auxiliary entity can fail due to any kind of failure. Along with the definition of *Auxiliary Entity Protection Set*, we define *Auxiliary Cumulative Fractional Minterm Hit Value* (ACFMHV) for designing the the heuristic. We first define *Auxiliary Fractional Minterm Hit Value* (AFMHV) which is used in defining ACFMHV.

Definition: The Auxiliary Fractional Minterm Hit Value of an IDR $D_j \in \mathcal{F}(E)$ is denoted by $AFMHV(D_j|E')$. It is calculated as $AFMHV(D_j|E') = \sum_{i=1}^{m} \frac{1}{|s_i|}$. Let x_j denote the entity in the right hand side of the IDR D_j. m denotes all the minterms in which the entity x_j appears over all IDRs. The parameter s_i denotes i^{th} such minterm with $|s_i|$ being its size. If an auxiliary entity is placed at D then the value computed above provides an estimate implicit impact on protection of other non operational entities.

Definition: The Auxiliary Cumulative Fractional Minterm Hit Value of an IDR $D_j \in \mathcal{F}(E)$ is denoted by $ACFMHV(D_j)$. It is computed as $ACFMHV(D_j) = \sum_{\forall x_i \in AP(D|E')} AFMHV(D_{x_i}|E')$ where D_{x_i} is the IDR for entity $x_i \in AP(D|E')$. The impact produced by the protected entities when IDR D is allocated with an auxiliary entity over set $A \cup B$ is implicitly provided by this definition.

The heuristic is provided in Algorithm 6. At any given iteration the auxiliary entity is placed at the IDR which protects the most number of entities from failure. In case of a tie the entity having highest ACFMHV value is chosen. At any given iteration the algorithm greedily maximize the number of entities protected from *induced*

Algorithm 6: Heuristic solution to AEAP

```
Data: A system $\mathcal{I}(E, \mathcal{F}(E))$, set of $E'$ entities failing initially, set $\mathcal{A}$ of auxiliary
      entities and budget $S$
. Result: Sets $D_{sol}$ and $P_f$
begin
    Initialize $D_{sol} = \emptyset$ and $P_f = \emptyset$;
    while $S \neq 0$ do
        For each IDR $D \in \mathcal{F}(E)$ compute the $AP(D|E')$ ;
        if Multiple IDRs have same highest cardinality $AP(D|E')$ then
            For each IDR $D \in \mathcal{F}(E)$ compute $ACFMHV(D)$ ;
            Let $D_p$ be an IDR having highest $ACFMHV(D_p)$ among all $D_i$'s in the
              set of IDRs having highest cardinality of the set $AP(D_i|E')$;
            Update $D_{sol} = D_{sol} \cup D_p$ and add an auxiliary entity from $\mathcal{A}$ as a
              disjunction to the IDR $D_p$;
            Update $P_f = P_f \cup AP(D_p)$, $S \leftarrow S - 1$ and $\mathcal{A}$ by removing the auxiliary
              entity added ;
        else
            Let $D_p$ be an IDR having highest cardinality of the set $D \in \mathcal{F}(E)$;
            Update $D_{sol} = D_{sol} \cup D_p$ and add an auxiliary entity from $\mathcal{A}$ as a
              disjunction to the IDR $D_p$;
            Update $P_f = P_f \cup AP(D_p|E')$, $S \leftarrow S - 1$ and $\mathcal{A}$ by removing the
              auxilary entity added ;
        Prune the system $\mathcal{I}(E, \mathcal{F}(E))$ by removing the IDRs for entities in
          $AP(D_p|E')$ and removing the same set of entities from $E$ ;
    return $D_{sol}$ and $P_f$ ;
```

failure. Algorithm 6 runs in polynomial time, more specifically the time complexity is $\mathcal{O}(Sn(n+m)^2)$ (where $n = |E|$ and $m =$ Number of minterms in $\mathcal{F}(E)$).

6.3 Experimental Results

To perform a comparative study of the heuristic with the ILP, we use the same data sets as outlined in Chapter 4. Additionally, the initial failure set is computed using $\mathcal{K}$ most vulnerable entities problem and the same value of $\mathcal{K}$ as in ENH problem are chosen for each data set (as in Table 4.4). 5 distinct allocation budgets were considered for each data set. The allocation budget for a given data set was chosen between $[1, 17]$. The set $\mathcal{A}$ containing auxiliary entities are chosen from the set of operational entities for a given data set. Figures 6.1 - 6.12 shows the comparison of the Heuristic solution with the ILP in terms of total number of entities protected for a given allocation budget. The run-time comparison of the solutions are provided in Table 6.3. A maximum percent difference of 13% (ILP compared with Heuristic) in the number of entities protected can be seen for 89 bus system when allocation budget is 1 (Figure 6.5). However, for most of the cases the heuristic produces near optimal or optimal solution. The heuristic also compute the solutions nearly 200 times faster than the ILP for larger systems as seen in Table 5.1. Hence it can be claimed that the heuristic solution to AEAP produces near optimal solution at a much faster time compared to the ILP solution.

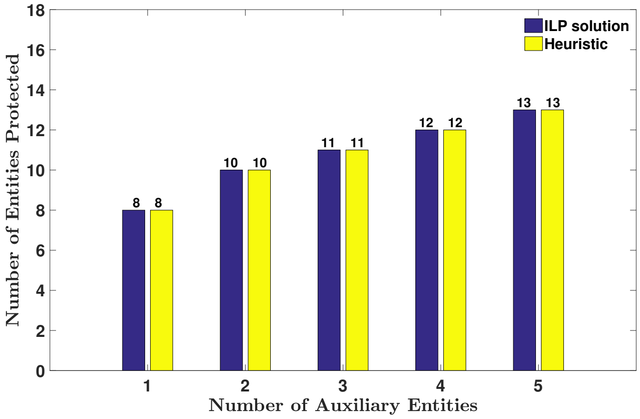

Figure 6.1: Comparison of ILP Solution with Heuristic for 24 Bus System (AEAP Problem)

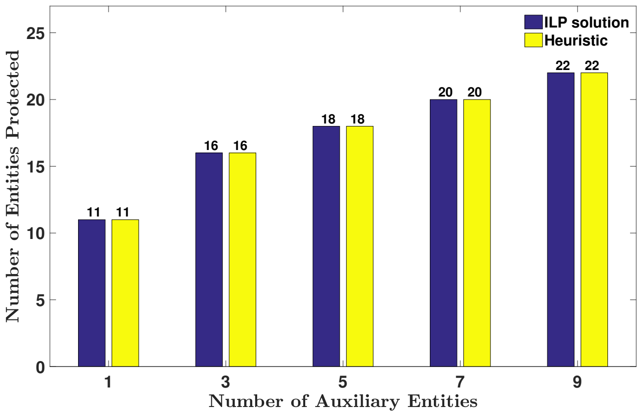

Figure 6.2: Comparison of ILP Solution with Heuristic for 30 Bus System (Auxiliary Entity Allocation Problem Problem)

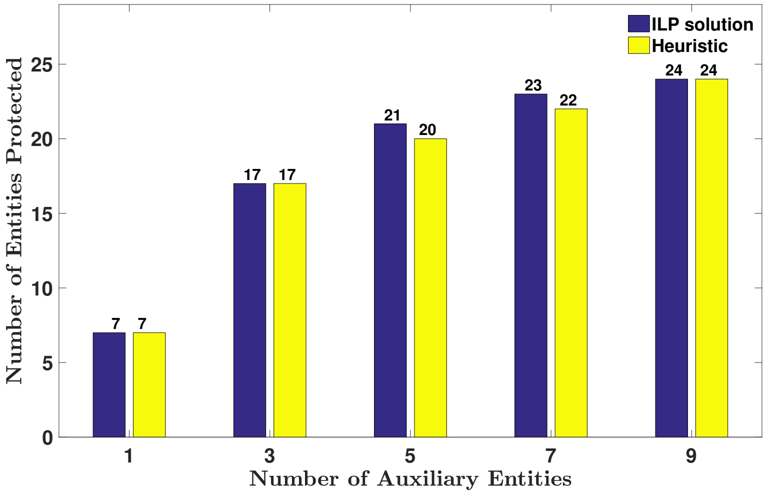

Figure 6.3: Comparison of ILP Solution with Heuristic for 39 Bus System (Auxiliary Entity Allocation Problem)

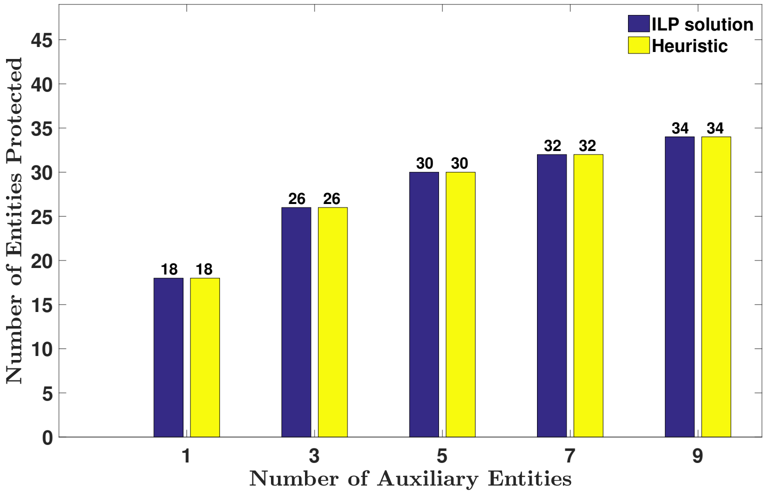

Figure 6.4: Comparison of ILP Solution with Heuristic for 57 Bus System (Auxiliary Entity Allocation Problem)

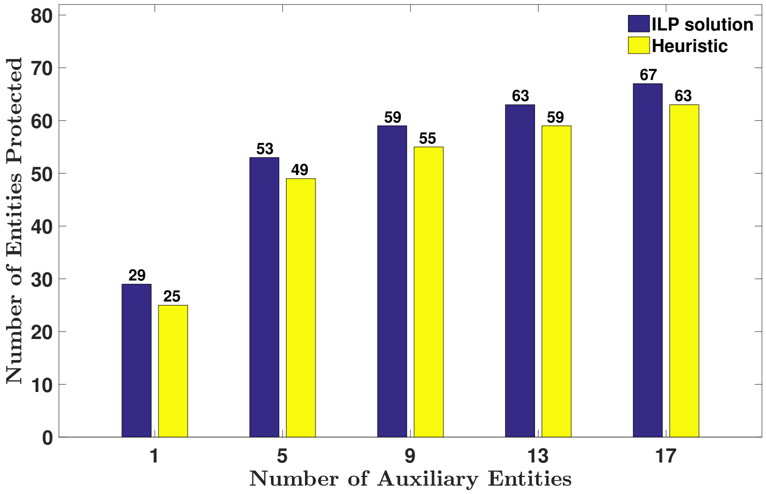

Figure 6.5: Comparison of ILP Solution with Heuristic for 89 Bus System (Auxiliary Entity Allocation Problem)

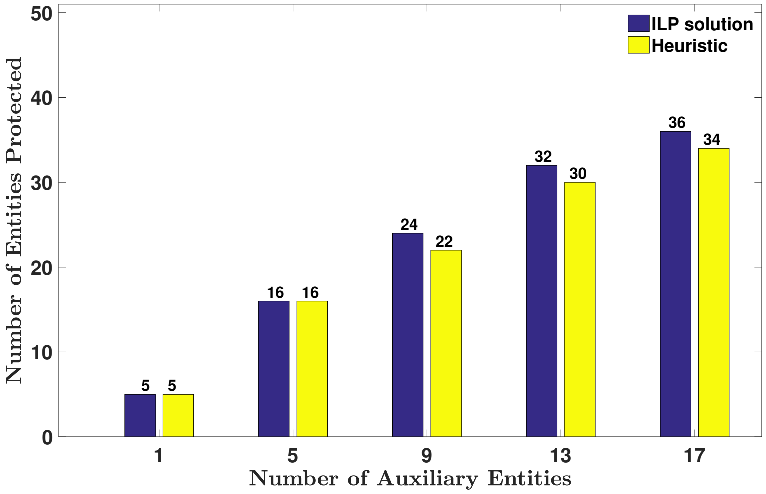

Figure 6.6: Comparison of ILP Solution with Heuristic for 118 Bus System (Auxiliary Entity Allocation Problem)

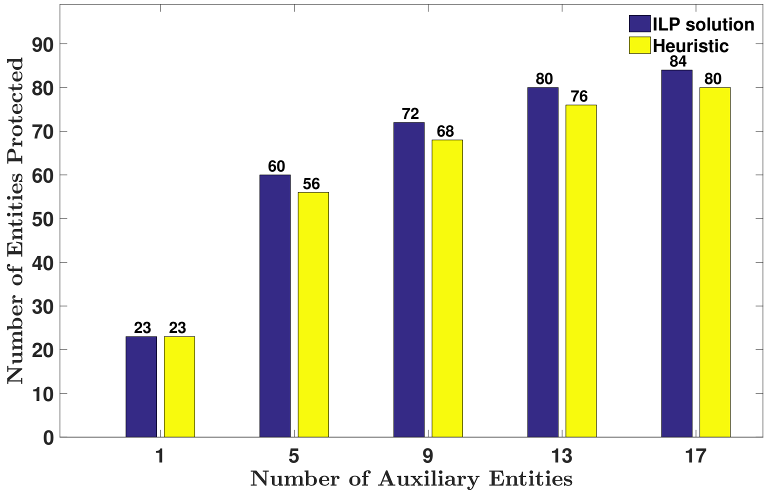

Figure 6.7: Comparison of ILP Solution with Heuristic for 145 Bus System (Auxiliary Entity Allocation Problem)

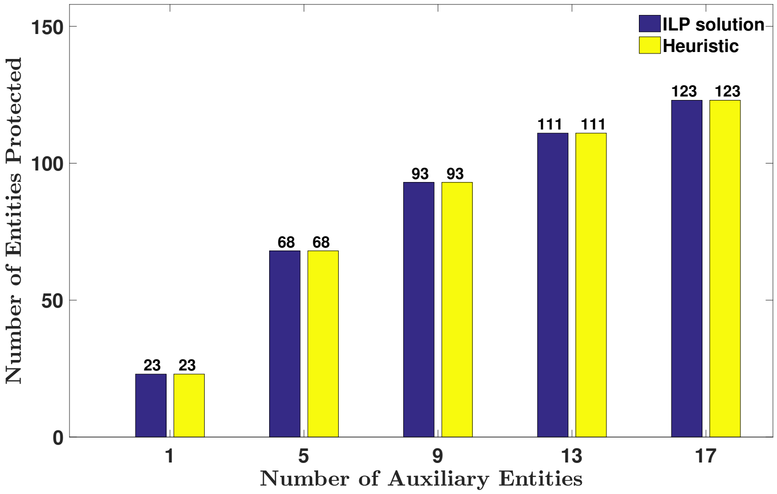

Figure 6.8: Comparison of ILP Solution with Heuristic for 300 Bus System (Auxiliary Entity Allocation Problem)

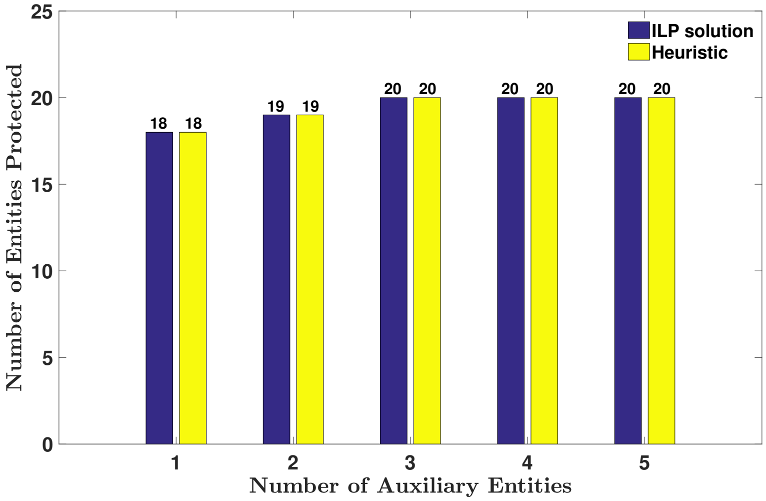

Figure 6.9: Comparison of ILP Solution with Heuristic for Region 1 (Auxiliary Entity Allocation Problem)

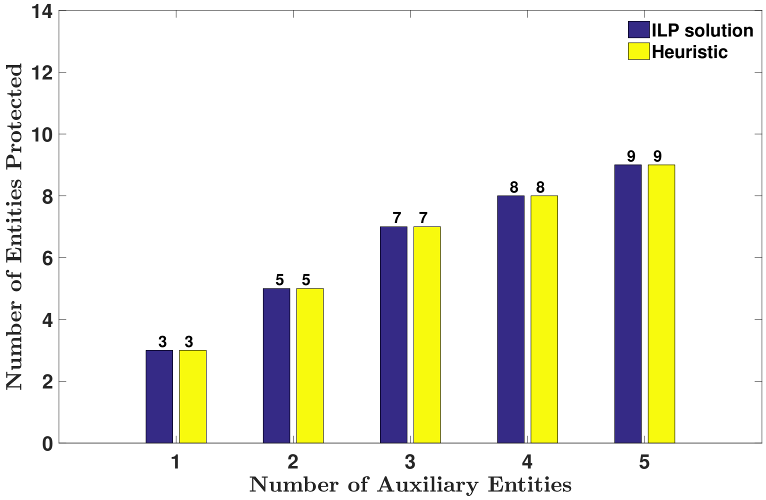

Figure 6.10: Comparison of ILP Solution with Heuristic for Region 2 (Auxiliary Entity Allocation Problem)

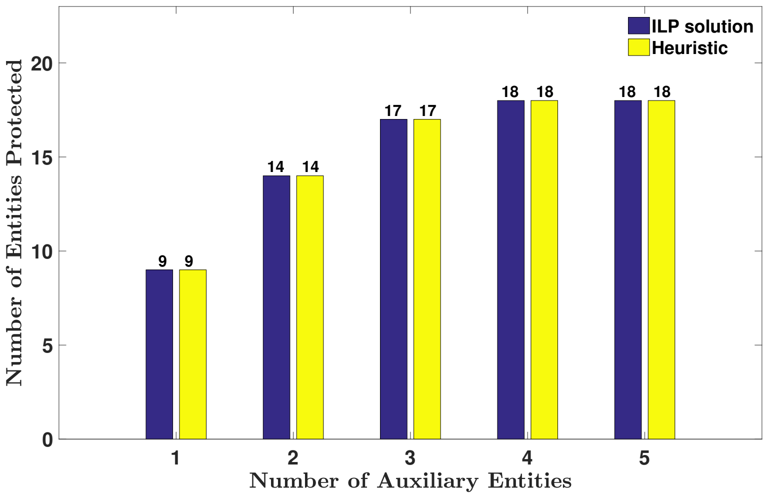

Figure 6.11: Comparison of ILP Solution with Heuristic for Region 3 (Auxiliary Entity Allocation Problem)

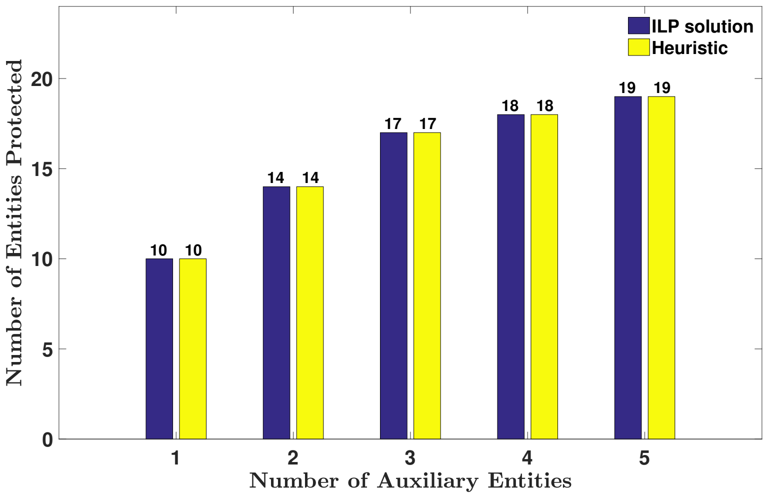

Figure 6.12: Comparison of ILP Solution with Heuristic for Region 4 (Auxiliary Entity Allocation Problem)

DataSet	**Running time (in sec)**									
	P1		**P2**		**P3**		**P4**		**P5**	
	ILP	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**
24 bus	0.49	0.01	0.20	0.01	0.20	0.01	0.21	0.01	0.18	0.01
30 bus	0.69	0.01	0.35	0.01	0.37	0.01	0.35	0.01	0.36	0.01
39 bus	0.9	0.01	0.49	0.01	0.46	0.01	0.45	0.01	0.46	0.01
57 bus	2.07	0.01	1.62	0.12	1.58	0.01	2.85	0.01	2.02	0.01
89 bus	16.5	0.06	14.6	0.03	14.8	0.03	14.1	0.02	14.9	0.02
118 bus	17.2	0.02	16.2	0.04	17.0	0.05	16.3	0.03	16.6	0.02
145 bus	79.1	0.03	76.7	0.06	76.9	0.05	76.3	0.03	76.2	0.04
300 bus	240	0.11	233	0.40	232	0.46	233	0.27	231	0.30
Region 1	0.57	0.01	0.35	0.01	0.34	0.01	0.34	0.01	0.35	0.01
Region 2	13.6	0.01	12.8	0.01	12.9	0.01	12.8	0.01	12.8	0.01
Region 3	1.50	0.01	1.29	0.01	1.24	0.01	1.37	0.01	1.27	0.01
Region 4	1.28	0.01	1.09	0.01	1.03	0.01	1.05	0.01	1.07	0.01

Table 6.3: Run Time Comparison of Integer Linear Program and Heuristic for Different Data Sets (Auxiliary Entity Allocation Problem)

Chapter 7

THE ROBUSTNESS PROBLEM

This chapter addresses a new metric for *Robustness*. We utilize IIM to model the dependencies and consider two type of dependencies while performing our experimental analysis — (a) interdependent power-communication network, (b) intra-dependent power network. It is critical to understand which set of entities which when failed initially would pose damage beyond a certain threshold. This understanding would provide the operator to employ proper protocols to prevent wide-scale failures. Using this motivation the chapter defines and provide solutions to compute Robustness. In IIM both the case (a) and (b) types of dependencies / inter-dependencies is generically represented as $\mathcal{I}(E, \mathcal{F}(E))$, where E are the set of entities and $\mathcal{F}(E)$ are the set of dependency relations portraying the dependencies in the infrastructure(s). The metric computing the Robustness is defined by using two parameters $K \in I^{+} \cup \{0\}$ and $\rho \in \mathbb{R}$ with $0 < \rho \leq 1$. If a minimum of $K+1$ entities need to fail for a failure of at least $\rho(|E|)$ entities then the system is (K, ρ)-robust. Utilization of this metric has the following advantages (a) For a new deployment of entities in an infrastructure, this metric provides the equipment installation personnel with information to make the system less vulnerable to initial failure triggers caused by human or nature, (b) For existing infrastructures the operator can use the metric to identify critical sections of the system based upon the extent of failure that section would cause.

7.1 Problem Formulation

We define a new metric for computing Robustness of a system $\mathcal{I}(E, \mathcal{F}(E))$. For a given system the metric is denoted by (K, ρ) where $K \in I^{+} \cup \{0\}$ is an integer and

$\rho \in \mathbb{R}$ is a real valued paramter with $0 < \rho \leq 1$. A system $\mathcal{I}(E, \mathcal{F}(E))$ is (K, ρ) robust if a minimum of $K+1$ entities need to fail initially for a final failure of at least $\rho|E|$ entities.

Power Network	**Comm. Network**
$a_1 \leftarrow b_2$	$b_1 \leftarrow a_1 a_2$
$a_2 \leftarrow b_1 + b_2$	$b_2 \leftarrow a_1 + a_3 a_4$
$a_3 \leftarrow b_2 b_3$	$b_3 \leftarrow a_2 a_3$
$a_4 \leftarrow b_1 b_3 + b_4$	$b_4 \leftarrow a_1$

Table 7.1: IDRs for the Constructed Example

Entities	Time Steps (t)						
	0	1	2	3	4	5	6
a_1	1	1	1	1	1	1	1
a_2	0	0	0	0	1	1	1
a_3	0	0	0	0	1	1	1
a_4	0	0	1	1	1	1	1
b_1	0	1	1	1	1	1	1
b_2	0	0	0	1	1	1	1
b_3	0	0	0	0	0	1	1
b_4	0	1	1	1	1	1	1

Table 7.2: Failure Cascade Propagation when Entity $\{a_1\}$ Fail at Time Step $t = 0$. A Value of 1 Denotes Entity Failure, and 0 Otherwise

Consider the system described in Table 7.1. It can be seen in Table 7.2 that the initial failure of the entity a_1 causes all the entities to fail in the steady state. Hence, a minimum of 1 entity needs to fail for failure of any number of entities ranging from 1 to 8 in the system. So for any ρ value the system is (0,ρ) robust. With this definition the primary challenge is to compute this robustness metric of a system. In our approach we take as input a system $\mathcal{I}(E, \mathcal{F}(E))$ and ρ with $0 < \rho \leq 1$ and compute the minimum number of entities (say K') require to fail at $t = 0$ that would cause a failure of at least $\rho|E|$ entities in total. With $K' = K - 1$ we then term the system as (K, ρ) robust. We term this as the *Robustness Computation* (RC) problem. A formal description of the decision and optimization version of the RC problem is stated below —

The Robustness Computation (RC) problem
Instance— *A system $\mathcal{I}(E, \mathcal{F}(E))$, an integer $K \in I^+$ and a real valued parameter $\rho \in \mathbb{R}$ with $0 < \rho \leq 1$.*
Decision Version— *Does there exist a set of entities $S_I \subseteq E$ and $|S_I| \leq K$ which when failed initially causes a final failure of at least $\rho|E|)$ entities.*
Optimization Version— *Find the minimum set of entities (say K') which when fail initially would cause a total final failure of at least $\rho|E|$ entities. The system would then be $(K' - 1, \rho)$ robust.*

It is to be noted that the decision version is developed as a negation to the RC problem i.e. a solution would ensure that the underlying system is **not** (K, ρ) robust. To find the solution to the RC problem the negation to the RC problem has to iterated from $K = |E| - 1$ to 0. Consider $K = K'$ as the first integer in the iteration for which there is a *no* answer to the negation of the RC problem. Then the robustness of the system is (K', ρ). So using a polynomial number of computation on the method computing the negation of the RC problem, the RC problem can be solved. Hence, the negation of the RC problem can be used to analyze the computational complexity of the RC problem.

7.2 Computational Complexity Analysis

The computational complexity analysis of the RC problem is described in this section. As noted earlier that analyzing computational complexity to the negation of the RC problem is same that of the RC problem. We denote the negation to RC problem as $\overline{RC}$. We prove that the decision version of the $\overline{RC}$ problem is NP-complete in Theorem 12. Additionally we analyze two sub-cases by imposing restrictions on the IDRs. In the first restricted case we provide a polynomial time solution to the RC problem. For the second restricted case we prove the $\overline{RC}$ problem to be NP-complete under the restriction and use the result to derive an in-approximability bound on the problem.

Theorem 12. *The decision version of the $\overline{RC}$ problem is NP-complete.*

Proof. We prove the NP-completeness by giving a transformation from the *Hitting Set Problem.* An instance of the hitting set problem consists of a set of elements S and a set $\mathcal{S} = \{S_1, S_2, S_3, .., S_n\}$ where $S_i \subseteq S$, $\forall S_i \in \mathcal{S}$. The question asked in the problem is given an integer M does there exist a set $S' \subseteq S$ with $|S'| \leq M$ such that each subset in $\mathcal{S}$ contains at least one element from S'. From an instance of the hitting set problem we create an instance of the $\overline{RC}$ problem as follows. Consider a system $\mathcal{I}(E, \mathcal{F}(E))$ with $E = A \cup B$. For each element $x_i \in S$ we add an entity b_i to set B. Similarly for each subset $S_i \in \mathcal{S}$ we add an entity a_i to set A. For each subset $S_i = \{x_m, x_n, x_p\}$ (say) we create an IDR $a_i \leftarrow b_m b_n b_p$. The value of K is set to M and ρ is set to $\frac{M+|\mathcal{S}|}{|S|+|\mathcal{S}|}$. It is to be noted that there wont be any cascading failure due to absence of dependency relations of B type entities.

Let there exists a solution to the hitting set problem. So each subset $S_i \in \mathcal{S}$ has at least one element from set S' (with $|S'| = M$). Hence killing the corresponding B type entities from the constructed instance would kill all A type entities. Thus the fraction of entities killed is $\frac{M+|\mathcal{S}|}{|S|+|\mathcal{S}|} = \rho$ solving the $\overline{RC}$ problem.

On the other way round let there exist a solution to the $\overline{RC}$ problem. It can be shown that the initial failure set would always be chosen from set B to fail $\rho = \frac{M+|\mathcal{S}|}{|S|+|\mathcal{S}|}$ fraction of entities. This is because failure of any A type entity cannot trigger failure of any other entity. Moreover the total number of entities in final failure set is $M + |\mathcal{S}|$ (as $|\mathcal{S}| = |A|$). Thus the failure set must contain all A type entities except for M other entities which has to be chosen from set B. So a solution to $\overline{RC}$ problem consisting of entities $B' \subseteq B$ would ensure that for each entity $a_i \in A$ at least one entity in its IDR is killed initially. So the set of elements in S' corresponding to the entities in B' would solve the hitting set problem. Hence proved □

7.2.1 Restricted Case I: Problem Instance with One Minterm of Size One

The IDRs in the set $\mathcal{F}(E)$ have minterms of size 1. With two entities e_i and e_j the IDR $e_i \leftarrow e_j$ represents this case. Additionally any entity can appear at most once on the left side of the IDR. We provide a polynomial time algorithm (Algorithm 7) and prove its optimality (Theorem 13) that solves the RC problem for Case I.

To develop the algorithm we use the definition of *Kill Set* of an entity $e_i \in E$ (as defined in Section 4.1). Using the concept of *Kill Set* Algorithm 7 is developed. For a given value of ρ the algorithm returns a set of entity R which when killed initially would cause failure of at least $\rho|E|$ entities. Theorem 13 proves that the set R returned has the minimum possible cardinality for a given value of ρ. Thus the system is $(|R| - 1, \rho)$ robust.

Algorithm 7: Algorithm solving RC problem optimaly for IDRs with Restricted Case I type dependencies

```
Data: A system $\mathcal{I}(E, \mathcal{F}(E))$ and a real valued parameter $\rho \in (0,1]$.
Result: A set of entities $R$ in $\mathcal{I}(E, \mathcal{F}(E))$.
begin
    For each entity $e_i \in (E)$ compute the set of kill sets $\mathcal{C} = \{C_{e_1}, C_{e_2}, ..., C_{e_{|E|}}\}$,
      where $C_{e_i} = KillSet(e_i)$ ;
    Initialize $\mathcal{D} \leftarrow \emptyset$ and $E \leftarrow \emptyset$ ;
    while $|\mathcal{D}| < \rho(|E|)$ do
        Let $e_j$ be the entity having highest $|C_{e_j}|$, in case of a tie choose arbitrarily ;
        Update $R \leftarrow R \cup \{e_j\}$;
        Update $\mathcal{D} \leftarrow \mathcal{D} \cup C_{e_j}$;
        for $(i = 1; i \leq |E|; i++)$ do
            $C_{e_i} \leftarrow C_{e_i} \backslash C_{e_j}$ ;
    return $E$ ;
```

Theorem 13. *Algorithm 7 solves the RC problem for Restricted Case I optimally in polynomial time.*

Proof. Computation of *Kill Sets* for all E entities can be done in $O((|E|)^3)$ [18]. The while loop runs for maximum of $|E|$ times when $\rho = 1$ and *Kill Set* of each entity is only composed of the entity itself. The highest cardinality *Kill Set* among all *Kill Sets* can be found in $O(|E|)$. The for loop iterates for $|E|$ times with computation inside it taking $O(|E|)$ time per iteration. Hence, the time complexity of the while loop in total is $O((|E|)^3)$. So the overall time complexity of Algorithm 7 is $O((|E|)^3)$.

We claim that Algorithm 7 returns the optimal value of robustness parameter $K = |R| - 1$ of an IDN $\mathcal{I}(E, \mathcal{F}(E))$ with set R containing the minimum number of entities that causes failure of at least $\rho(|E|)$ entities. The claim is proved by contradiction. Let R_{OPT} be the optimal set that causes failure of at least $\rho(|E|)$ entities and e_n be an entity in $R_{OPT} \backslash R$. It is proved in [18] that in this restricted case for any two entities e_i and e_j, $C_{e_i} \cap C_{e_j} = \emptyset$ or $C_{e_i} \cap C_{e_j} = C_{e_i}$ or $C_{e_i} \cap C_{e_j} = C_{e_j}$ where $e_i \neq e_j$. At any iteration of the while loop the entity e_j with highest cardinality *Kill Set* is selected. Inside the for loop all entities having $C_{e_i} \cap C_{e_j} = C_{e_i}$ and the entity itself would have its *Kill Set* updated to $\emptyset$. Hence the *Kill Set* of the entity x_n would either be set to $\emptyset$ at some iteration of the while loop or didn't have the highest cardinality at any iteration. Hence adding e_n to optimal solution would have made no difference or reduce the number of failed entities. Hence a contradiction. So, Algorithm 7 returns the minimum number of entities that causes failure of at least $\rho(|E|)$ entities. □

7.2.2 Restricted Case II: Problem Instance with an Arbitrary Number of Minterms of Size One

This restricted case is composed of IDRs having arbitrary number of minterms of size 1. The IDRs of this case can be represented as $e_i \leftarrow \sum_{q=1}^{p} e_q$. The given example has p minterms each of size 1. Thus to kill e_i, all entities in its IDR must be non-operational. In Theorem 14 we prove that the decision version of the $\overline{RC}$ problem for this restricted case is NP complete. Using the instance creation as described in Theorem 14 and the NP-completeness proff we provide an in-approximability bound on the $\overline{RC}$ problem in Theorem 15.

Theorem 14. *The decision version of the $\overline{RC}$ problem for Case III is NP-complete.*

Proof. We prove that the problem is NP-complete by giving a reduction from the *Densest p-Subhypergraph* problem [32], a known NP- complete problem. An instance of the Densest p-Subhypergraph problem includes a hypergraph $G = (V, E_V)$, a parameter p and a parameter M. The problem asks the question whether there exists a set of vertices $|V'| \subseteq V$ and $|V'| \leq p$ such that the subgraph induced with this set of vertices has at least M completely covered hyperedges. From an instance of the Densest p-Subhypergraph problem we create an instance of the $\overline{RC}$ problem as follows. Consider a system $\mathcal{I}(E, \mathcal{F}(E))$ with $E = A \cup B$. For each vertex $v_i \in V$ we add an entity b_i to set B. Similarly, for each hyperedge $e_j \in E$ we add an entity a_j to set A. For each hyperedge e_j with $e_j = \{v_m, v_n, v_q\}$ (say) an IDR of form $a_j \leftarrow b_m + b_n + b_q$ is created and added to $\mathcal{F}(E)$. The value of K is set to p and ρ is set to $\frac{p+M}{|V|+|E_V|}$. It is to be noted that there wont be any cascading failure due to absence of dependency relations of B type entities.

Let there exist a solution to the Densest p-Subhypergraph problem. Then there exist a set $V' \subseteq V$ and $|V'| = p$ that covers completely at least M hypedges in E_V. Thus killing the B type entities corresponding to the vertices in V' would cause at least M A type entities to fail. Hence the fraction of entities killed is $\geq \frac{p+M}{|V|+|E_V|} = \rho$. So the solution of the Densest p-Subhypergraph problem solves the $\overline{RC}$ problem.

For the created instance of the $\overline{RC}$ problem all entities in set B can only fail initially. The A type entities can either fail initially or through induced failure of failing B type entities. Hence initial failure of entities from set B would have the most impact on final number of entities failed. Let us assume that there exists one or many solutions to the $\overline{RC}$ problem. Then at least one solution would have entities only from set B. For this solution the number of entities killed on initial failure of p B type entities is at least $p + M$. The additional M entities killed belongs to set A. So the vertices in V corresponding to the entities in B would completely cover at least M hyperedges. Thus the solution of $\overline{RC}$ problem solves the Densest p-Subhypergraph problem. Hence proved. □

Theorem 15. *The $\overline{RC}$ problem is hard to approximate within a factor $\frac{1}{2^{log(n)^{\lambda}}}$ (where $n = |E|$) for some $\lambda > 0$.*

Proof. In [32] it is proved the Densest p-Subhypergraph problem is hard to approximate within a factor of $\frac{1}{2^{log(n)^{\lambda}}}$ with $\lambda > 0$. For IDRs of form in this restricted case it is shown in Theorem 14 that Densest p-Subhypergraph problem is a special case of the $\overline{RC}$ problem. So this in-approximability bound holds for the $\overline{RC}$ problem as well. Hence proved. □

7.3 Solutions to the RC Problem

Owing to the problem being NP-complete we first propose an Integer Linear program (ILP) that solves the problem optimally. Since, the run time of the ILP becomes exponential with input size so a sub-optimal heuristic that runs in polynomial time is also proposed in this section.

7.3.1 Optimal Solution for the RC problem

For a given parameter $\rho \in (0, 1]$ and a system $\mathcal{I}(E, \mathcal{F}(E))$, we formulate an ILP that computes the minimum number of entities which need to fail at $t = 0$ for a final failure of $\rho(|E|)$ entities. Let K' denote the solution to the ILP. *The system is then (K, ρ) robust with $K = K' - 1$.* For each entity where $e_i \in E$ a set of variables x_{id} $(0 \leq d \leq |E| - 1)$ are created in the ILP. d is the parameter which denotes the time step. If x_{id} =1 then the entity e_i is non-operational at time setp d and operational if $x_{id} = 0$. state. Using these variable creations the objective of the ILP is provided in Equation 7.1

$$min \sum_{i=1}^{|E|} x_{i0} \tag{7.1}$$

The constraints of the ILP are formally described as follows:

Constraint Set 1: $x_{id} \geq x_{i(d-1)}, \forall d, 1 \leq d \leq |E| - 1$ The constraint ensures that if an entity e_i fails at time step d, it should remain in a state of failure for all subsequent time steps.

Constraint Set 2: Consider an IDR of form $e_i \leftarrow e_j + e_k e_l + e_m e_n e_q$. A set of constraints is developed for each such IDR that captures the cascading failure process. The set of constraints are described below —

Step 1: We bring in new variables to denote minterms of size greater than one. In this example, two new variables c_1 and c_2 are introduced to represent the minterms $e_k e_l$ and $e_m e_n e_q$ respectively. This is equivalent of adding two new IDRs $c_1 \leftarrow e_k e_l$ and $c_2 \leftarrow e_m e_n e_q$ with the transformed IDR being $e_i \leftarrow e_j + c_1 + c_2$.

Step 2: For each IDR corresponding to the c type variables, we establish a linear constraint to capture the failure propagation. For an IDR $c_2 \leftarrow e_m e_n e_q$ the constraint is represented as $c_{2d} \leq x_{m(d-1)} + x_{n(d-1)} + x_{q(d-1)}, \forall d, 1 \leq d \leq |E| - 1$.

Step 3: Similarly, for each transformed IDR, we introduce a linear constraint to capture the failure propagation. For an IDR $e_i \leftarrow e_j + c_1 + c_2$ the constraint is represented as $N \times x_{id} \leq x_{j(d-1)} + c_{1(d-1)} + c_{2(d-1)}, \forall d, 1 \leq d \leq |E| - 1$. Here N is the number of minterms in the IDR (in this example $N = 3$).

Constraint Set 3: It must also be satisfied that at time step $|E| - 1$, at least $\rho(|E|)$ entities fail in total. This can be captured by introducing the constraint $\sum_{i=1}^{m} e_{i(t_f)} \geq \rho(|A| + |B|)$.

With the objective in (7.1) and set of constraints, the ILP finds the minimum number of entities K' which when failed initially, causes at least $\rho(|E|)$ entities to fail in total at $t = |E| - 1$. Thus using this solution the system is $(K' - 1, \rho)$ robust.

7.3.2 Heuristic Solution for the RC problem

A sub-optimal heuristic solution to the RC problem is proposed in this section. The heuristic utilizes the definition of *Kill Set* (as defined in Section 4.1) and Cumulative Fractional Minterm Hit Value of an entity (defined below). Using these definitions the heuristic is provided in Algorithm 8.

Definition: *Fractional Minterm Hit Value:* For an entity $e_j \in E$ in a dependent system $\mathcal{I}(E, \mathcal{F}(E))$ the Fractional Minterm Hit Value is denoted as $FMHV(e_j, X)$. It is calculated as $FMHV(e_j, X) = \sum_{i=1}^{m} \frac{1}{|s_i|}$. In the formulation m are the minterms in which e_j appears over all IDRs except for entities in set X. The parameter s_i denotes i^{th} such minterm. If the entity e_j is killed then the computed value provides an estimate of the future impact on killing of other operational entities.

Definition: *Cumulative Fractional Minterm Hit Value:* The Cumulative Fractional Minterm Hit Value of an entity $e_j \in E$ is denoted as $CFMHV(e_j)$. It is computed as $CFMHV(e_j) = \sum_{\forall e_i \in KS(e_j)} FMHV(e_j, KS(e_j))$. This gives a measure of the future impact on killing functional entities when the entity e_j is killed.

In Algorithm 8, for each iteration of the while loop the operational entity having highest cardinality Kill Set is selected. This ensures that at each step the number of entities failed is maximized. In case of a tie, the entity having highest cardinality Cumulative Fractional Minterm Hit Value among the set of tied entities is selected. This causes the selection of the entity that has the potential to kill maximum number of entities in the subsequent steps. Thus, the heuristic greedily minimizes the set of

Algorithm 8: Heuristic Solution to RC problem

```
Data: A system $\mathcal{I}(E, \mathcal{F}(E))$ and a real valued parameter $\rho \in (0, 1]$.
Result: An integer $|K_H| - 1$ where $K_H$ is a set of entities that when killed initially
        fails at least $\rho(|E|)$ entities
begin
    Initialize $\mathcal{D} \leftarrow \emptyset$ and $K_H \leftarrow \emptyset$ ;
    while $|\mathcal{D}| < \rho(|E|)$ do
        For each entity $e_i \in E \backslash D$ compute the kill set $C_{e_i}$;
        For each entity $e_i \in E \backslash D$ compute $CFMHV(e_i)$;
        Let $e_j$ be the entity having highest $|C_{e_j}|$ ;
        if There exists multiple entities having highest cardinality Kill Set then
            Let $e_p$ be an entity having highest $CFMHV(e_p)$ with $e_p$ in the set of
              entities having highest cardinality Kill Set;
            If there is a tie choose arbitrarily;
            Update $K_H \leftarrow K_H \cup \{e_p\}$ ;
            Update $\mathcal{D} \leftarrow \mathcal{D} \cup C_{e_p}$;
            Update all dependencies in $\mathcal{F}(E)$ by removing entities in the left and
              right side of the IDRs that belong to $C_{e_p}$;
        else
            Update $K_H \leftarrow K_H \cup \{e_j\}$ ;
            Update $\mathcal{D} \leftarrow \mathcal{D} \cup C_{x_j}$;
            Update all dependencies in $\mathcal{F}(E)$ by removing entities in the left and
              right side of the IDRs that belong to $C_{e_p}$;
    return $|K_H| - 1$ ;
```

entities which when killed initially fails at least ρ fraction of total entities in the IDN. The heuristic overestimates the parameter K while determining the robustness (K, ρ)

of an IDN. The value of the parameter K is equal to $|K_H| - 1$ which is the output of Algorithm 8. Algorithm 8 runs in polynomial time, more specifically the run time is $\rho n(n+m)^2$ (where $n = |E|$ and $m =$ Number of minterms in $\mathcal{F}(E)$).

7.3.3 Comparative Study of the ILP and Heuristic for the Problems

We perform a comparative study of the ILP with the heuristic for the RC problem using the same data sets as used in ENH problem (Chapter 4). We compared the heuristic solution with the ILP by considering 5 different values of $\rho = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Table 7.3 compares the quality of the solution for different values of ρ and data sets. The quality of the solution is measured as the K value of the robustness metric returned for the corresponding value of ρ. The run time performance of the solutions are enumerated in Table 7.4. For a ρ value of 0.7 in Region 2 it can be seen that the K value in heuristic is 50% of the optimal. This is maximum percent difference in the quality of the solution across all the experiments. However, it can be seen that for most of the cases the heuristic performs very close to the optimal solution produced by the ILP. Moreover, from Table 7.4 it can be inferred that the heuristic solves the RC problem much faster than the ILP with heuristic performing nearly 100 times faster for systems having large number of entities. Hence, using this results in can be reasonably argued that the heuristic solution provides near optimal solution in much lesser computation and hence can be used to solve the RC problem.

DataSet	**ρ and Robustness Values**									
	0.1		**0.3**		**0.5**		**0.7**		**0.9**	
	ILP	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**
24 bus	2	2	6	6	15	15	27	27	39	39
30 bus	1	1	4	4	12	12	26	26	40	40
39 bus	1	1	4	4	17	18	34	35	51	52
57 bus	1	1	9	10	26	27	53	54	80	81
89 bus	4	5	19	23	78	82	137	141	196	200
118 bus	7	8	35	35	89	89	148	148	208	208
145 bus	1	1	78	81	191	194	304	307	418	421
300 bus	5	5	30	33	145	148	287	290	429	432
Region 1	1	1	2	2	4	7	13	16	23	26
Region 2	1	1	4	4	7	9	12	18	21	27
Region 3	1	1	3	3	5	8	13	17	23	27
Region 4	1	1	3	3	7	9	16	20	25	30

Table 7.3: Quality of Solution Comparison of Integer Linear Program and Heuristic for Different Data Sets and Varying ρ (Robustness Problem)

DataSet	**ρ values and running time (in sec)**									
	0.1		**0.3**		**0.5**		**0.7**		**0.9**	
	ILP	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**	**ILP**	**Heu**
24 bus	0.25	0.01	0.20	0.01	0.19	0.01	0.22	0.01	0.31	0.01
30 bus	0.27	0.01	0.53	0.01	0.47	0.01	0.44	0.01	0.44	0.01
39 bus	0.45	0.01	0.63	0.01	0.90	0.01	0.76	0.01	0.75	0.01
57 bus	2.23	0.02	2.96	0.01	2.40	0.03	2.52	0.02	1.83	0.01
89 bus	12.5	0.12	20.1	0.10	16.0	0.20	13.7	0.13	9.87	0.09
118 bus	19.1	0.19	20.5	0.24	18.6	0.32	19.7	0.37	17.6	0.22
145 bus	81.3	0.17	90.3	0.88	87.8	0.81	86.1	0.65	81.9	0.71
300 bus	272	0.86	271	2.12	264	3.33	234	3.00	235	2.90
Region 1	0.48	0.01	0.65	0.01	0.53	0.01	0.32	0.01	0.36	0.01
Region 2	0.49	0.01	5.04	0.01	8.73	0.01	.38	0.01	0.24	0.01
Region 3	0.46	0.01	2.23	0.01	1.37	0.01	0.29	0.01	0.275	0.01
Region 4	0.50	0.01	4.97	0.01	2.31	0.01	0.47	0.01	0.31	0.01
Region 5	0.61	0.01	2.26	0.01	7.51	0.01	0.31	0.01	0.20	0.01

Table 7.4: Run Time Comparison of Integer Linear Program and Heuristic for Different Data Sets and Varying ρ (Robustness Problem)

Chapter 8

THE MIIR MODEL AND K CONTINGENCY LIST PROBLEM

The biggest challenge with the IIM formulation is the identification of the dependency relations between the different entities. This is done based on information obtained from subject matter experts. However, it has now become clear that this is not a very reliable procedure. The second problem is that IIM operates on Boolean logic, implying that the different entities can only have two values — 0 or 1, representing the state of the entity being operational or non-operational. However, this does not provide information about entities which are operating at near-failure state. For example, consider a line that is carrying 95% of its rated capacity. IIM only provides the state of the line (operational or non-operational). Hence the operator won't be alerted even though the line is reaching its peak carrying capacity (which might eventually cause it to fail). Thus, such scenarios limit the applicability of IIM. In order to overcome the limitations of IIM and to extend the application domain of IDRs for long-term planning and short-term operational management, the Multi-scale Implicative Interdependency Relation (MIIR) model is proposed. The MIIR model uses the notion of IDRs with added features to capture the power flow in transmission lines/transformers and demand/generation of buses. Phasor Measurement Unit (PMU) data can be used to generate the dependency equations as well as obtaining power flow and demand/generation values for actual systems. Using the MIIR model we study the K contingency list problem in this chapter. At a given time t the problem solves for a set of K components in the power network which when made non-operational at time t would cause the maximum number of healthy entities to fail. Additionally, the solution would provide insights into components which are operating at their near capacity limits. Such a solution would have an immediate benefit to a system operator making decisions in real-time to prevent large-scale power failures.

In this chapter, the MIIR model is developed based on the power network. It is aimed to have a near accurate abstraction of the power flow dynamics and capture cascading failure propagation in the same. It is to be noted with proper modification the model has the potential to be extended for performing a similar analysis in different inter/intra-dependent critical infrastructure system(s).

8.1 Model Variables

We consider load buses, generator buses, neutral buses and transmission lines / transformers as different types of entities. Let $E = \{e_1, e_2, \ldots, e_n\}$ denote the set of entities in the power network. Each entity $e_i \in E$ has three values associated with it — (i) a lower bound $e_{i,l}$, (ii) an upper bound $e_{i,u}$, and (iii) the instantaneous power value $e_{i,c,t}$ at time t of the entity. For a transmission line/transformer type entity e_k, the value of $e_{k,c,t}$ provides the power flow in that line at time t. Corresponding, for a load bus e_m and generator bus e_n the values $e_{m,c,t}$ and $e_{n,c,t}$ provides the load demand and generating value at time t. For the power transmission system, PMU data can be used to obtain the instantaneous power value $e_{i,c,t}$ of the entity e_i. The values of $e_{i,l}$ and $e_{i,u}$ can be easily obtained from the entity rating data. For a given time t the state of the entity is still Boolean (operational or not operational) and is guided by the following two factors — (a) $e_{i,c,t}$ satisfies the property $e_{i,l} < e_{i,c,t} < e_{i,u}$, (b) the corresponding dependency equation of e_i at time t is satisfied. Hence, if e_i has an IDR $e_i \leftarrow e_j \cdot e_k + e_l$ then for e_i to be operational both the properties — (a) (e_i and e_k) or e_l is operational at $t - \delta$, and (b) $e_{i,t} \leq e_{i,c,t} \leq e_{i,u}$ has to be satisfied. Here δ refers to the time within which the effect of failure of an entity is propagated to its dependent entity. So using MIIR model the power network at time

t is mathematically represented as $P(E, B, C_t, F)$ where E is the set of of entities, B is a set of tuples $\{e_{i,l}, e_{i,u}\}$ $(\forall e_i \in E)$ denoting the power value bound on the entity, C_t consist of instantaneous power value $e_{i,c,t}$ $(\forall e_i \in E)$ at time t and F contains the set of dependency equations for the entities in E. A similar notation has been used in [33] but our notation brings out a completely different topological aspect of the power network.

8.2 Generating and Obtaining the Model Variables for a Power Network

We illustrate our strategy to generate the dependency equations F and the set C_t of a power network $P(E, B, C_t, F)$ at a given time t. The MATPOWER [31] software is used to generate the simulated data. For a given time t and a standard bus system (containing a set of buses and transmission lines/transformers E), the software uses load demand of the bus, the impedance of the transmission lines/transformers, etc. to solve the power flow. The software produces the voltage of each bus in the system as the output. The software suite also includes a wide range of test systems along with power ratings of the components for all such systems. We restrict ourselves to analyze the real power flow. Firstly, for a given solution, we formally state the procedure to obtain the tuple values of the set B and instantaneous power value contained in the set C for generator buses, load buses, neutral buses and transmission lines/transformers —

- *Generator Bus:* The real part of the power generated is taken as the value of $e_{a,c,t}$ for a generator bus $e_a \in E$. The upper bound $e_{a,u}$ is set to its real generation capacity (supplied in the MATPOWER suite) and the lower bound $e_{a,l}$ is set to 0. It is to be noted that some generator buses have load demand. Consider e_x be a generator bus with load demand d units and real instantaneous power generated $e_{x,c,t}$ units. Without any loss of generality, such a bus is split into a generator bus e_{x1} with 0 load demand (instantaneous power generated $e_{x1,c,t}$) and a load bus e_{x2} with instantaneous load demand d units (instantaneous power generated 0). A transmission line e_{x12} is constructed that connects e_{x1} to e_{x2} with an instantaneous power flow of d units flowing from e_{x1} to e_{x2}.

- *Load Bus:* The real part of the load demand is taken as instantaneous demand value $e_{b,c,t}$ of a load bus $e_b \in E$. For a load bus e_b, both its upper and lower bound is set to the instantaneous demand value $e_{b,c,t}$. Essentially, our assumption is that a load bus does not change its demand value irrespective of any failure.

- *Neutral Bus:* For a neutral bus $e_d \in E$ the values of $e_{d,l}, e_{d,u}$ and $e_{d,c,t}$ are set to 0.

- *Transmission Lines/Transformers:* For two buses e_1 and e_2 connected by a transmission line/transformer e_{12} the power flowing through the transmission line/transformer is calculated as $P_{12} = Real(V_1 * (\frac{V_1 - V_2}{I_{12}})^*)$, where V_1 is the voltage at bus e_1, V_2 is the voltage at bus e_2 (V_1 and V_2 returned by the MATPOWER solver) and I_{12} is the impedance of the transmission line/transformer e_{12} (obtained from the supplied bus system file of MATPOWER). P_{12} is the real component of the power flowing in the transmission line/transformer e_{12}. The lower bound is set to 0 and the upper bound is taken as the rated capacity of the transmission line/transformer. The instantaneous power value $e_{12,c,t}$ is set to $|P_{12}|$ (absolute value).

The description of this system indicates that power flows from bus e_1 to e_2 if P_{12} is positive and vice-versa otherwise. As a result, we can interpret the direction of power flow in the line from the solution which we obtained from MATPOWER. We use this solution to generate the set F which is the set of dependency equations. As an example, we consider the nine bus system which is shown in Figure 3.4 (refer to Chapter 3). This figure describes a power network $P(E, B, C_t, F)$, at time instance t with E being the set of entities containing generator buses from G_1 to G_3, load buses L_1 to L_4, neutral buses $\{N_1, N_2\}$ and transmission lines/transformers T_1 through T_9. The figure also provides the instantaneous power values by solving the power network flow based on demand/generation at some time instant t. The red blocks denote the instantaneous real power generated by a generator, the green blocks denote instantaneous real load demands and the blue nodes are neutral. The values in the grey blocks denote the flow of power in the transmission lines/transformers with the arrows denoting the direction of the power flow. There aren't any IDRs for the transmission lines. Consider a bus b_1 connected to buses b_2, b_3 through transmission lines/transformers b_{12} (between b_1 and b_2) and b_{13} between (b_1 and b_3). Also, let power flow from buses b_2, b_3 to bus b_1. The dependency equation for the bus b_1 is constructed as disjunction of minterms of size 2 (this consists of the bus from which the power is flowing and the respective transmission line/transformer) with each disjunction corresponding to buses from which power is flowing to it. For this example the dependency equation $b_1 \leftarrow b_{12}b_2 + b_{13}b_3$ is created. Using this definition, the dependency equations for the buses in Figure 3.4 are as follows — (a) $L_1 \leftarrow T_1 \cdot G_1$, (b) $L_2 \leftarrow T_2 \cdot L_1 + T_7 \cdot N_2$, (c) $L_3 \leftarrow T_3 \cdot L_1 + T_4 \cdot N_1$, (d) $L_4 \leftarrow T_6 \cdot N_1 + T_8 \cdot N_2$, (e) $N_1 \leftarrow T_5 \cdot G_3$, (f) $N_2 \leftarrow T_9 \cdot G_2$.

8.3 Dynamics of the MIIR model

To understand the dynamics of cascading failure in power network based on the MIIR model we first create the abstract representation $P(E, B, C_0, F)$ (which is constructed using the technique discussed in Section 8.2) for a power network at time

$t = 0$. An event of initial failure is assumed to occur at time $t = 0$ with failure cascade propagating in unit time steps. For an entity that is operational at time step $t = \tau$ the following equations are required to be satisfied —

$$\sum_{e_m \in O_{e_g}} e_{m,c,\tau} = \sum_{e_n \in I_{e_g}} e_{n,c,\tau} + e_{g,c,\tau}, \forall e_g \in G \tag{8.1}$$

$$\sum_{e_m \in O_{e_l}} e_{m,c,\tau} = \sum_{e_n \in I_{e_l}} e_{n,c,\tau} - e_{l,c,\tau}, \forall e_l \in L \tag{8.2}$$

$$\sum_{e_m \in O_{e_k}} e_{m,c,\tau} = \sum_{e_n \in I_{e_k}} e_{n,c,\tau} + e_{k,c,\tau}, \forall e_k \in N \tag{8.3}$$

Equations (8.1)-(8.3) dictate the law of conservation of energy for each bus in the system. That is, we assume that the power flowing out from a bus is equal to the power flowing into it for a unit time step.

In (8.1), the lines through which power flows out and into the generator bus e_g (where $G \subset E$ contains all generator buses) are represented by sets I_{e_g} and O_{e_g} respectively. Equations (8.2) and (8.3) uses the same notations for load bus e_l (where $L \subset E$ contains all load buses) and neutral bus e_k (where $N \subset E$ contains all neutral buses), respectively. In (8.3), the value of $e_{k,c,t} = 0$ for all time steps and hence it can be simplified as $\sum_{e_m \in O_{e_k}} e_{m,c,t} = \sum_{e_n \in I_{e_k}} e_{n,c,t}$. Additionally, for a generator bus there is no power injected to it. Hence 8.1 can be re-written as $\sum_{e_m \in I_{e_g}} e_{m,c,t} = e_{g,c,t}, \forall e_g \in G$. We use this abstract representation of the power network i.e. $P(E, B, C_0, F)$ at time $t = 0$. Using this the cascading failure process of the power network on an event of initial failure of $E' \subset E$ at time step $t = 0$ is detailed out in Algorithm 9.

In Algorithm 9, at every iteration the flow values are adjusted based on Equations 8.1-8.3 at line 8 and entities are made non-operational based on the two conditions mentioned at lines 7 and 9. The cascading process continues if new entities fail in the previous time step (condition $size \neq |S|$). As evident we assume that there is a

Algorithm 9: Algorithm describing the failure cascading process in power network using MIIR model

```
Data: A power network $P(E, B, C_0, F)$ at time $t = 0$ and a set of initially failing
      entities $E' \subset E$
Result: A set of failed entities $S$
begin
    Initialize $S \leftarrow E'$, $size \leftarrow 0$ ;
    Increment $t \leftarrow t + 1$ ;
    while $size \neq |S|$ do
        Set $size \leftarrow |S|$;
        Remove all entities whose dependency equations are not satisfied and add
          them to set $S$;
        Adjust power flow values $e_{k,c,t}$ of transmission line/transformer entities $e_k$
          and generating values $e_{g,c,t}$ of generator buses $e_g$ such that Equations 8.1
          -8.3 are satisfied.;
        Remove all entities whose bounds are not satisfied and add them to set $S$ ;
        Increment $t \leftarrow t + 1$ ;
    return $S$ ;
```

unit time delay for an entity to become non operational if its dependency equations are unsatisfied. All entities whose bound values are not satisfied are made non-operational at that time step after power flow calculation. It is to be noted that the dependency equations are generated from a graph which is *directed acyclic*. Owing to this property the cascade reaches a steady state (no new entities are non-operational) within $\mathcal{O}(|E|)$ time steps. This can be explained as follows. Consider a single initial failure of an entity. If no entity fails the cascading algorithm would continue till at most $|E| - 1$ time steps since the maximum distance between two nodes in a directed acyclic graph is $|E| - 1$ (considering each edge having a weight 1). If more than one entity fails then the cascade is expected to stop before $|E| - 1$. Hence the number of cascading time steps is strictly upper bounded by $|E| - 1$.

Algorithm 9 assumes that there exist a method to compute the flow value equations to get the instantaneous power values of the entities. This is equivalent to computing the AC power flow equations again (using MATPOWER) which would be time intensive and does not make use of the abstraction created by the MIIR model for fast decision-making. Moreover, using general graph theoretical algorithmic techniques might result in multiple solutions of instantaneous power values when solving a given set of power flow equations thus resulting in ambiguity. To counteract this, in our abstraction, we use the notion of **Worst-Case Cascade Propagation** (WCCP) in Algorithm 9. Qualitatively, the instantaneous value of power flows and power generator at every time step $t > 0$ of the cascade is set to a value that would cause the maximum number of entities to fail at the end of the cascade. Computation of this power flow values using WCCP is proved to be NP-complete in Section 8.5. We devise a mixed integer program to get the optimal solution and a greedy heuristic to get a sub-optimal solution in polynomial time in Section 6.2.

8.4 Case Study: The 2011 Southwest Blackout

In this subsection the performance of MIIR with WCCP is tested on a real power system event: the 2011 Southwest Blackout. All data used in this analysis are obtained from [2]. An abstraction of the Southwest Power System is provided in Figure 8.1. The abbreviations used in Figure 8.1 are — Western Electricity Coordinating Council (WECC), Serrano (SE), Devers (DE), San Onofre Nuclear Generating Station (SONGS), San Diego Gas & Electric (SDG&E), Miguel (MI), Imperial Valley (IV), Imperial Irrigation District (IID), Comision Federal de Electricidad's (CFE, corresponding to Baja California Control Area), North Gila (NG), Hassayampa (HA), Palo Verde (PV), and Western Area Power Administration-Lower Colorado (WAPA). The blue, orange and green blocks in Figure 8.1 represents neutral, load and generator buses respectively. The transmission lines/transformers are labeled $T1-T17$ with the arrows indicating the directions of the pre-disturbance power flows. On September 8, 2011, an initial trip of the HA-NG transmission line ($T11$) caused blackout in SDG & E region. The objective here is to verify whether MIIR model with WCCP is able to capture the power outage.

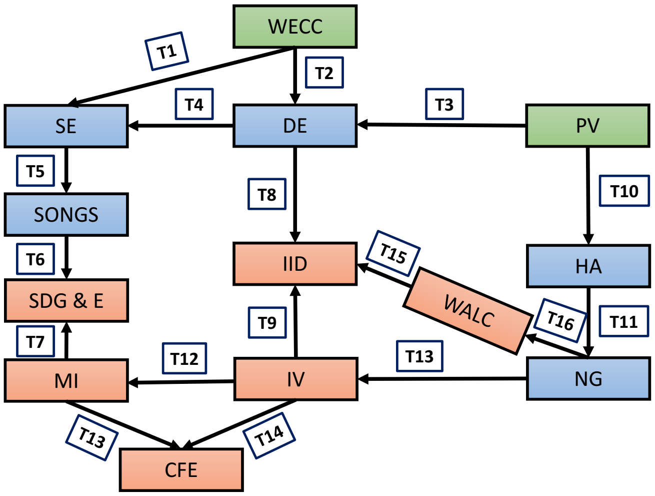

Figure 8.1: An Abstraction of the Southwest Power System.

The dependency equations in Table 8.1 without the bounds and instantaneous power values of the entities (buses and transmission lines) corresponds to the set F. Consider tripping of the entity $T11$ at $t = 0$. Just considering the IDRs itself, the component NG fails at $t = 1$, $WALC$ and IV at $t = 2$, CFE and MI at $t = 3$. The pre-disturbance load demands of SDG&E and IID were approximately 5000 MW and 900 MW, respectively, while the generation bounds on PV and WECC were [0, 4000 MW] and [0, 10000 MW], respectively. After failure of T11, SDG&E and IID would try to meet their bulk load demands through the generator buses PV and WECC via

Dependency Equations
$SE \leftarrow T1 \cdot WECC + T4 \cdot DE$
$DE \leftarrow T3 \cdot PV + T2 \cdot WECC$
$SONGS \leftarrow T5 \cdot SE$
$SDG\&E \leftarrow T6 \cdot SONGS + T7 \cdot MI$
$IID \leftarrow T8 \cdot DE + T15 \cdot WALC + T9 \cdot IV$
$MI \leftarrow T12 \cdot IV$
$IV \leftarrow T13 \cdot NG$
$CFE \leftarrow T14 \cdot IV + T13 \cdot MI$
$WALC \leftarrow T16 \cdot NG$
$NG \leftarrow T11 \cdot HA$
$HA \leftarrow T10 \cdot PV$

Table 8.1: IDRs of the Southwest Power System

T6 and T8. The bound on T6 is [0, 2200 MW] and T8 is [0, 1800 MW]. Both PV and WECC have enough generation capacity to meet the load demand of SDG&E and IID. At $t = 3$ owing to the load demand of SDG&E the transmission line $T6$ would have try to have a power flow of 5000 MW instantly. Thus $T6$ would trip at $t = 3$ causing SDG&E to trip at $t = 4$. Owing to this the power flowing through $T1, T4$ and $T5$ would reduce down to 0 at $t = 4$. The power flow in $T8$ would increase to 900 MW at $t = 3$ for supplying power to IID. Thus the steady state is reached at $t = 4$ and MIIR model accurately predicts the blackout of SDG&E region.

8.5 K Contingency List — Problem Formulation

It is important from a power system operator's point of view to understand and know the most critical entities in the network at a given time. This would enable the operator to make more reliable decisions when unforeseen events/failures occur. For larger systems, an automation that provides the operator with this information would be highly beneficial. Owing to this we develop the K Contingency List (KCoL) problem using MIIR model with WCCP. For a given time t and an integer K the problem provides the operator with a list of K entities which when failed initially causes the maximum number of entities to fail at the steady state of cascade propagation. Qualitatively, for a given integer K the problem finds a set E' ($|E'| = K$) entities which when failed initially maximizes the total number of entities failed at the end of the cascading process. A formal description of the KCoL problem using WCCP with MIIR model for the Power Network is provided —

Input: (a) A power network $P(E, B, C_t, F)$ where $E = G \cup L \cup N \cup T$. Set of entities G, L, N and T are disjoint and contains the generator buses, load buses, neutral buses and transmission lines/transformers, respectively. (b) two positive integers K and S.

Decision Version: Does there exist a set of K entities in E whose failure at time t would result in a failure of at least S entities in total at the end of the cascading process?

Optimization Version: Compute the a set of K entities in a power network $P(E, B, C_t, F)$ whose failure at time t would maximize the number of entities failed at the steady state of cascade propagation.

We prove the problem is NP-complete to solve in Theorem 16

Theorem 16. *The KCoL problem using MIIRA model is NP-complete.*

Proof. The problem is proved to NP-complete by a reduction from the densest $p-$subhypergraph problem [32]. An instance of the densest $p-$subhypergraph problem consists of a hypergraph $H = (V, E)$ and two parameters p and M. The decision version of the problem finds the answer to *whether there exists a set of vertices $V' \subset V$ and $|V'| \leq p$ which completely covers at least M hyper-edges.*

From an instance of the densest $p-$subhypergraph we create an instance of the KCoL problem as follows. We start with an empty set of entities G, L and T and an empty set F that would comprise of the dependency equations. A load type entity L_j is added to set L for each hyper-edge $E_j \in E$ with instantaneous load demand $L_{i,c,t}$ set to the number of vertices that comprise this hyper-edge. For each vertex $V_i \in V$ we add a generator type entity G_i to set G. The upper bound on the capacity of the generator G_i is set to the sum of all instantaneous load demands $L_{i,c,t} + 1$ for which the corresponding hyper-edge E_j contains the vertex V_i. For each hyper-edge E_j consisting of vertices V_x, V_y, V_z (say) three transmission line type entities T_x, T_y and T_z are added to set T and a dependency equation $L_j \leftarrow T_x \cdot G_x + T_y \cdot G_y + T_z \cdot G_z$ is created and added to set F. The upper bound of the transmission line is set to the load demand $+1$ of the entity it connects to (e.g., in this case, the maximum capacity of each transmission line T_x, T_y, T_z are set to the instantaneous load demand $L_{i,c,t} + 1$). The parameter S of KCoL problem is set to $p + M$ and K is set to p (i.e. p entities fail at time t). Thus the created instance satisfy the property of the graph from which the dependency relations are computed being *Directed Acyclic*. In the initial operating condition at time t, all transmission lines have a line flow value of 1 unit with each generator G_i producing P_i units of power, where P_i is the number of load entities it is connected to. Hence all load demands are satisfied. It can be directly followed that an instance of KCoL problem can be created from an instance of densest $p-$subhypergraph problem in polynomial time.

It is to be noted that for the created instance — (1) Each transmission line has the capacity to satisfy the complete load demand of the load type entity it is connecting, (ii) Each generator has the capacity to satisfy the load demand of all the load type entities it is connected to. Hence an initial failure of one or more entities would not cause any transmission line or generator to trip (fail) because of exceeding its maximum capacity. Thus the generators and transmission lines are susceptible only to initial failure whereas the load entities are vulnerable to both initial and induced failures. However, failure of load entities can not cause any induced failure. Induced failure of the load entity can be caused only when each minterm in its dependency equation have at least one failed entity. Thus no entity fail due to change in power flow values.

Now consider there exist a solution to the densest $p-$subhypergraph problem. Hence there exist a set of p vertices V' that completely covers M hyper-edges. Failing the generator type entities corresponding to the vertices in V' would thus fail at least M load entities at $t+1$ according to the instance construction. Thus a total of at least $p+M$ entities would fail which solves the KCoL problem. On the other way round consider there exist a solution to the KCoL problem. As reasoned earlier, a load entity cannot cause any induced failure. Hence if a load entity is in the solution then it can be substituted with any operational generator entity without loss of correctness. Similarly, if a transmission line type entity is in the solution it can be replaced by a generator type entity it is connected to. Using this substitution a solution thus comprises of entities $G' \subset G$. All M (or greater than M) entities that fail due to the initial failure of p entities belongs to set L. Thus the substituted solution (or original solution if no substitution is required) would consist of generator type entities that cause failure of these M (or greater than M) load entities. Hence selecting the vertices corresponding to G' would ensure that at least M hyper-edges are completely covered solving the densest $p-$subhypergraph problem. Hence proved.

□

8.6 Solutions to the Problem

Owing to the KCoL problem being NP-complete, we obtain the optimal solution using Mixed Integer Program (MIP). However, as we require to compute the contingency list fast, we also devise a polynomial time heuristic that provides a sub-optimal solution to the problem.

8.6.1 Optimal Solution using Mixed Integer Program (MIP)

As a reference frame, we consider that the initial failure occurs at time step $t = 0$. It is shown in Section 8.3 that the number of time steps in the cascade is upper bounded by $|E| - 1$. We devise an MIP that solves the KCoL problem optimally for a power network $P(E, B, C_0, F)$ (the abstraction constructed for $t = 0$). Irrespective of whether the steady state is reached before or at time step $|E| - 1$, in our MIP we try to maximize the number of entities failed at $t = |E| - 1$ when K entities fail at $t = 0$. Moreover, it can not be predicted when the cascading failure stops. Hence, the MIP is bound to check for solution to compute the maximum number of entities that can fail till the maximum possible time step, i.e. $|E| - 1$. Firstly, the list of variables used in the MIP formulation are discussed—

- *Variable List 1:* For each entity $e_i \in E$ a variable set $x_{i,t}, \forall t, 0 \leq t \leq |E| - 1$ are created. The value of $x_{i,t}$ is 0 if the entity is operational at time step t and 1 otherwise.

- *Variable List 2:* For each entity $e_i \in E$ a variable set $y_{i,t}, \forall t, 0 \leq t \leq |E|-1$ is created. From the set C_0 we can get the initial instantaneous power value $e_{i,c,0}$ of an entity e_i. The value of $y_{i,0}$ is set to $e_{i,c,0}$. All the instantaneous values are real thus comprising the set of non integer variables in the program.

Using these definitions and the list of variables created, the objective of the MIP is provided in (8.4) and the constraints of the MIP are formally described.

$$max \sum_{i=1}^{|E|} x_{i,|E|-1} \tag{8.4}$$

Subjected to:

Constraint Set 1: $\sum_{i=1}^{|E|} x_{i,0} = K$. This constraint sets the number of entities failed at time step $t = 0$ to K.

Constraint Set 2: $x_{i,d} \geq x_{i,t-1}, \forall t, 1 \leq t \leq |E|-1$. This ensures that an entity that is not operational at time step $t = d$ would remain non-operational in all times step $t > d$.

Constraint Set 3: Consider an IDR of form $e_i \leftarrow e_a \cdot e_b + e_c \cdot e_d$. To capture the cascading failure process, a set of constraints is developed and described below —

Step 1: New variables are introduced to represent the minterms. In this example, two new variables c_{ab} and c_{cd} are created to represent the terms $e_a \cdot e_b$ and $e_c \cdot e_d$. This is equivalent of adding two new IDRs $c_{ab} \leftarrow e_a \cdot e_b$ and $c_{cd} \leftarrow e_c \cdot e_d$ with the transformed IDR being $e_i \leftarrow c_{ab} + c_{cd}$.

Step 2: A linear constraint is developed for the c type variables to capture the failure propagation. For an IDR $c_{ab} \leftarrow e_a \cdot e_b$, the constraint is represented as $c_{ab,t} \leq$

$x_{a,t-1} + x_{b,t-1}, \forall t, 1 \leq t \leq |E| - 1$. This captures the condition that $c_{ab,t}$ is equal to 1 only if at least one of the entities e_a or e_b is non operational.

Step 3: For each transformed IDR a linear constraint is introduced. For an IDR $e_i \leftarrow c_{ab} + c_{cd}$ the constraint is represented as $N \times x_{i,t} \leq c_{ab,t-1} + c_{cd,t-1}, \forall t, 1 \leq t \leq |E| - 1$. Here N is the number of minterms in the IDR (in this example $N = 2$).

Constraint Set 4: For a given load bus entity e_l, the constraint $y_{l,t} = 0, \forall t, 0 \leq t \leq |E| - 1$ is added denoting that the instantaneous power demand of all the load bus remain constant at each time step. Similarly, for a given neutral bus entity e_n, the constraint $y_{n,t} = 0, \forall t, 0 \leq t \leq |E| - 1$ is added.

Constraint Set 5: For a given generator bus entity e_p and transmission line entity e_q, the constraints $x_{p,t} \leq \frac{y_{p,t}}{e_{p,u}}, \forall t, 1 \leq t \leq |E| - 1$ and $x_{q,t} \leq \frac{y_{q,t}}{e_{q,u}}, \forall t, 1 \leq t \leq |E| - 1$ are added. As this is a maximization problem, the x type variable of the corresponding generator/transmission line entity would be set to 1 when it operates beyond its rated upper bound. The constraints $y_{a,t} \geq 0$ and $y_{a,t} \leq e_{a,u} + 1$ are added at all time steps for each generator or transmission line type entity e_a. This limits the maximum value of these entities to its upper bound plus one and them failing only if their instantaneous power value is just above the upper bound.

Constraint Set 6: To capture the power flow equations given by (8.1)-(8.3) the following constraints are developed. Consider the equation $\sum_{e_m \in O_{e_l}} e_{m,c,t} = \sum_{e_n \in I_{e_l}} e_{n,c,t} - e_{l,c,t}$. Naively, this can be constructed as a non-linear constraint $\sum_{e_m \in O_{e_l}} (1 - x_{m,t}) \times y_{m,t} = \sum_{e_n \in I_{e_l}} (1 - x_{n,t}) \times y_{n,t} - (1 - x_{l,t}) y_{l,t+1}$. The constraint denotes that the instantaneous flow values of the different power network entities are taken into consideration if the the load bus is operational at the next time step (as failure due to IDR is reflected after 1 unit of time). This constraint can be linearized as $\sum_{e_m \in O_{e_l}}(y_{m,t} - x_{m,t} \times e_{m,u}) = \sum_{e_n \in I_{e_l}}(y_{n,t} - x_{n,t} \times e_{n,u}) - (y_{l,t} - x_{l,t+1}e_{l,u})$. If a transmission line/transformer e_n fails at time instant t then its instantaneous power value is set to its upper bound (owing to constraint set 5). This would equate the term $(y_{n,t} - x_{n,t} \times e_{n,u})$ corresponding to this transmission line/transformer to 0. If the transmission line/transformer e_n is operational then $x_{n,t} = 0$ and hence $(y_{n,t} - x_{n,t} \times e_{n,u})$ would equate to $y_{n,t-1}$ thus being considered in the power flow equation. Similarly if the load bus is not operational the value of $(y_{l,t} - x_{l,t+1}e_{l,u})$ is set to 0. These constraints are constructed for all time steps $0 \leq t \leq |E| - 2$ and similar constraints are generated for equations 8.1 and 8.3 as well.

Constraint Set 7: For each transmission line/transformer type entity $e_a \in E$ flowing out power from a bus type entity e_b the constraint $x_{a,t} \leq x_{b,t}$ is added for each time step $1 \leq t \leq |E| - 1$. This captures the condition that if a bus type entity fails then all transmission lines/transformers to which it transmits power also fails.

It is to be noted that there won't be any infeasibility in solution arising due to the constraints. The load and neutral buses can only be made non-operational through their dependency equations. Whereas, the transmission lines/transformers and generators can be only made non-operational through change in power flow/generation values (as they don't have any dependency equations). The objective in (8.4) along with these set of constraints, finds the the set of K entities whose initial failure at $t = 0$ maximizes the number of entities failed at the end of the cascading process. As this is a maximization problem the power flow and generation at each time step is set to values that maximize the total number of entities failing at the steady state. Hence the MIP captures the notion of WCCP.

8.6.2 Heuristic Solution

In this section we design a sub-optimal heuristic that finds a solution to the KCoL problem in polynomial time. Primarily we use the definition of Kill Set defined in section 4.1 and Fractional Minterm Hit Value defined below.

Definition: *Fractional Minterm Hit Value:* For an entity $e_j \in E$ in an power network $P(E, B, C_t, F)$ the Fractional Minterm Hit Value is denoted as $FMHV(e_j)$. It is calculated as $FMHV(e_j) = \sum_{i=1}^{m} \frac{c_i}{|s_i|}$ where m are the minterms and for a given minterm m_i, c_i are the number of entities that belong to $KS(e_j)$ and $|s_i|$ is its size. This metric provides an estimate of impact of other operational entities that can be made non-operational at future time steps if the entity e_j is made non-operational.

Algorithm 10 returns a sub-optimal value of E' which when failed initially would greedily maximize (based on *Kill Set* and *FMHV*) the number of entities failed at the end of the cascade. The algorithm runs in $\mathcal{O}(Kn(n+m)^2)$ where $n = |E|$ and m are the total number of minterms. It is to be noted that the greedy failure maximization is done based on IDR. To get the actual number of entities failed when the set of entities E' fail initially we use the MIP. Essentially, we modify the constraint 1 such that only entities in E' fail at $t = 0$ and see the number of entities failed at the final time step. This gives us a measure to compare the efficacy of the heuristic solution with respect to the MIP.

8.7 Experimental Results

We analyzed the run time performance and quality of the heuristic solution with respect to MIP for different test systems. The quality of the solution is defined by the number of components reported to be non-operational for a given value of K. Specifically we used the $9, 14, 24, 30, 39, 57, 118, 145, 300$ and 2383 Winter Polish bus systems

```
Algorithm 10: Heuristic Solution to KoCL problem
Data: A power network $P(E, B, C_0, F)$ at time $t = 0$ and an integer $K$.
Result: A set of initially failing entities $E' \subset E$ and $|E'| \leq K$
begin
    Initialize $\mathcal{D} \leftarrow \emptyset$ $E' \leftarrow \emptyset$ and $K_H \leftarrow 0$;
    while $K_H < K$ do
        Update $K_H \leftarrow K_H + 1$;
        For each entity $e_i \in E \backslash \mathcal{D}$ compute the kill set $C_{e_i}$;
        For each entity $e_i \in E \backslash \mathcal{D}$ compute $FMHV(e_i)$;
        Let $e_j$ be the entity having highest $|C_{e_j}|$ ;
        if There exists multiple entities having highest cardinality Kill Set then
            Let $e_p$ be an entity having highest $FMHV(e_p)$ with $e_p$ in the set of
              entities having highest cardinality Kill Set;
            If there is a tie choose arbitrarily;
            Update $E' \leftarrow E' \cup \{e_p\}$, $\mathcal{D} \leftarrow \mathcal{D} \cup C_{e_p}$ ;
            Update all dependencies in $F$ by removing entities in the left and right
              side of the IDRs that belong to $C_{e_p}$;
        else
            Update $E' \leftarrow E' \cup \{e_j\}$, $\mathcal{D} \leftarrow \mathcal{D} \cup C_{x_j}$ ;
            Update all dependencies in $F$ by removing entities in the left and right
              side of the IDRs that belong to $C_{e_p}$;
    return $E'$ ;
```

available in MATPOWER. For a given test system, we used the MATPOWER AC power solver. Using the data, the abstract power network $P(E, B, C_0, F)$ was generated. On the constructed power network the MIP and heuristic solutions were executed. The implementation was done in Java and a student licensed version of

IBM CPLEX optimizer was used to solve the MIP. A UNIX system with 8 GB of RAM and intel i5 processor was used for the execution.

In Table 8.2, a comparison between the MIP and the heuristic solution with respect to the number of entities in non-operational state for different bus systems with K varied from 1 to 5 in steps of 1 are provided. Additionally, the total number of entities (buses and transmission lines) for each bus system is mentioned. It is to be noted that the total number of entities and the number of entities in non-operational state include the entities constructed for generator buses with non-zero load demand (as mentioned in Section 8.2). Table 8.3 reports the IDR generation time for each bus system along with the time taken to execute the MIP and heuristic for different values of K. Some insightful observations from the results are as follows — (a) The heuristic solution performs very nearly to that of MIP with respect to quality and have an almost same performance for $K = 1$. (b) For almost all the cases, the maximum percent difference in the number of non-operational entities in heuristic with respect to MIP is under 1% with a maximum percent difference of 7% for 57 bus system at $K = 3$. (c) It is observed that more than 50% of the total entities in a given test system will be non-operational if $K = 1$. This implies that the power system is extremely vulnerable even if a single entity is attacked, (d) For almost all the test systems from 9 to 300 the heuristic finds a solution to the *KCol* problem nearly 100 times faster than the MIP, (e) for the 2383 Winter Polish bus system the heuristic is 10 to 20 times faster. However, it is be noted that the comparison is done based on a serialized implementation of the heuristic and can be made faster by parallelization.

Hence, it can be reasonably argued that the Heuristic solves the *KCol* problem achieving near optimal solution at a much faster time compared to MIP. Thus the abstraction provided by the MIIR model along with the Heuristic solution can be used by a Power Network operator to obtain the K *Contingency List* in real-time.

DataSet	**Number of Entities**	*K* values and number of entities dead in MIP and heuristic									
		1		**2**		**3**		**4**		**5**	
		MIP	**Heu**	**MIP**	**Heu**	**MIP**	**Heu**	**MIP**	**Heu**	**MIP**	**Heu**
9 bus	24	15	15	19	19	24	24	24	24	24	24
14 bus	44	29	29	32	31	36	35	42	41	44	43
24 bus	80	48	48	51	51	54	54	58	58	61	61
30 bus	83	54	54	58	58	64	63	71	70	74	73
39 bus	105	63	63	70	70	74	74	79	78	84	82
57 bus	149	96	96	113	108	128	119	136	128	140	136
118 bus	405	240	240	245	245	250	250	255	254	260	260
145 bus	666	499	499	511	511	517	516	522	517	527	523
300 bus	847	493	491	506	506	513	513	524	524	531	530
2383wp bus	5923	3249	3249	3296	3278	3326	3308	3354	3332	3382	3353

Table 8.2: Quality of Solution Comparison of Mixed Integer Program and Heuristic for Different Power Network Bus Systems and Varying K

DataSet	**IDR Gen.**	*K* **values and running time in sec for MIP and in ms for Heuristic**									
		1		**2**		**3**		**4**		**5**	
	Time (ms)	**MIP**	**Heu**	**MIP**	**Heu**	**MIP**	**Heu**	**MIP**	**Heu**	**MIP**	**Heu**
9 bus	0.31	0.026	2	0.13	1	0.02	1	0.02	1	0.01	1
14 bus	0.34	0.09	2	0.14	1	0.08	1	0.05	1	0.04	1
24 bus	0.60	0.27	4	0.16	4	0.19	6	0.18	3	0.27	3
30 bus	0.55	0.24	5	0.20	5	0.17	6	0.16	3	0.18	4
39 bus	0.62	0.30	8	0.17	14	0.28	6	0.28	7	0.25	5
57 bus	0.76	1.44	17	0.65	13	0.60	15	0.74	15	0.60	17
118 bus	1.49	1.78	33	1.09	52	1.16	64	0.73	33	1.02	41
145 bus	2.22	10.48	60	9.45	83	9.49	91	10.26	147	12.72	76
300 bus	2.09	11.09	146	4.23	249	4.68	364	3.63	253	4.34	293
2383wp	7.66	315	5123	220	7323	225	11431	218	14749	262	18318

Table 8.3: Run Time Comparison of Mixed Integer Program and Heuristic for Different Power Network Bus Systems and Varying K

Chapter 9

CONCLUSION AND FUTURE DIRECTIONS

In concluding remarks this dissertation attempts to perform an in depth study on understanding inter/intra dependent critical infrastructure systems and analyzing their vulnerability along with approaches to protect them from such vulnerabilities. The limitations of the existing models in capturing the dependencies in critical infrastructure system is brought upon through a survey. For addressing these limitations the Implicative Interdependency Model is introduced. Different problems addressing vulnerability and protection analysis are proposed and solved using the IIM model. Owing to certain limitations of the IIM model, the MIIR model is proposed to counteract them. Using the MIIR model the K contingency list problem is formulated and solved that addresses vulnerabilities in power network. All the problems discussed in this dissertation are NP-complete. The polynomial time solutions derived for these problems are seen to be efficient in terms of quality and time performance when compared to Integer Linear programs (Mixed Integer program for the K contingency list problem) used to find the optimal solution.

This dissertation is a basis for a multitude of future research problems. Apart from this being a documentation, all source code and test data sets are made open source to support any future research. The online repository containing source codes to the solutions of the problems can be found in *https://github.com/jbanerje1989*. Some of the future research directions are discussed —

- *Some other problems that can be solved:* The IIM/MIIR model can be used to address some other pertinent problems in critical infrastructure system. The following can be of interest as future research problems — (a) *Analysis with incomplete/incorrect information*: Inaccurate/Incomplete dependency equations or other model parameters directly impact solutions of different problems. Such inaccuracies can arise due to many reasons such as incorrect data or approximations done due to unavailability of data. Thus studying the impact on the solutions of different problems under such inaccuracies is a possible research problem, (b) *Islanding in Multi-Layer Networks:* The concept of islanding is well studied in power network. Using the IIM model or the MIIR model this can be extended to multi-layer interdependent power communication network. The main task to address such a problem is to have a concrete and realistic definition of islanding in multi-layered network under the model setting.

- *Extending solutions to the problems using MIIR model:* The problems that are solved using the IIM model (i.e. Entity Hardening Problem, Targeted Hardening Problem, Auxiliary Entity Allocation Problem, Robustness Computation problem) can be extended and solved using the MIIR model.

- *Extending MIIR model to have an abstraction of interdependent systems:* The model is developed based on intra-dependencies in power network. It can be extended to have an accurate abstraction of interdependent power-communication network. Such an extension might require some changes in the model parameters and dynamics.

- *Building a toolkit and a GUI:* The ultimate research goal is to build a graphical user interface which a system operator can use to analyze vulnerability and make appropriate control actions to reduce failure in intra/inter dependent systems. A suite can be developed for different real world test beds that displays fast solutions to different problems using the MIIR model.

REFERENCES

[1] G Andersson, P Donalek, R Farmer, N Hatziargyriou, I Kamwa, P Kundur, N Martins, J Paserba, P Pourbeik, J Sanchez-Gasca, et al. Causes of the 2003 major grid blackouts in north america and europe, and recommended means to improve system dynamic performance. *Power Systems, IEEE Transactions on*, 20(4):1922–1928, 2005.

[2] Federal Energy Regulation Commission and the North American Electric Reliability Corporation. Arizona - Southern California Outages on September 8, 2011, Causes and Recommendations. Technical report, 04 2012.

[3] Yong Tang, Guangquan Bu, and Jun Yi. Analysis and lessons of the blackout in indian power grid on july 30 and 31, 2012. In *Zhongguo Dianji Gongcheng Xuebao(Proceedings of the Chinese Society of Electrical Engineering)*, volume 32, pages 167–174. Chinese Society for Electrical Engineering, 2012.

[4] William J. Clinton. Executive order eo 13010 critical infrastructure protection. Technical report, July 1996.

[5] Steven M Rinaldi. Modeling and simulating critical infrastructures and their interdependencies. In *System sciences, 2004. Proceedings of the 37th annual Hawaii international conference on*, pages 8–pp. IEEE, 2004.

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

[7] Peter Pederson, D Dudenhoeffer, Steven Hartley, and May Permann. Critical infrastructure interdependency modeling: a survey of us and international research. *Idaho National Laboratory*, pages 1–20, 2006.

[8] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, 2010.

[9] Jia Shao, Sergey V Buldyrev, Shlomo Havlin, and H Eugene Stanley. Cascade of failures in coupled network systems with multiple support-dependence relations. *Physical Review E*, 83(3):036116, 2011.

[10] Hua Lin, Santhoshkumar Sambamoorthy, Sandeep Shukla, James Thorp, and Lamine Mili. Power system and communication network co-simulation for smart grid applications. In *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, pages 1–6. IEEE, 2011.

[11] Hua Lin, Santhosh S Veda, Sandeep S Shukla, Lamine Mili, and James Thorp. Geco: Global event-driven co-simulation framework for interconnected power system and communication network. *IEEE Transactions on Smart Grid*, 3(3):1444–1456, 2012.

[12] Pengcheng Zhang, Srinivas Peeta, and Terry Friesz. Dynamic game theoretic model of multi-layer infrastructure networks. *Networks and Spatial Economics*, 5(2):147–178, 2005.

[13] Jianxi Gao, Sergey V Buldyrev, H Eugene Stanley, and Shlomo Havlin. Networks formed from interdependent networks. *Nature Physics*, 8(1):40–48, 2011.

[14] Vittorio Rosato, L Issacharoff, F Tiriticco, Sandro Meloni, S Porcellinis, and Roberto Setola. Modelling interdependent infrastructures using interacting dynamical models. *International Journal of Critical Infrastructures*, 4(1):63–79, 2008.

[15] Marzieh Parandehgheibi and Eytan Modiano. Robustness of interdependent networks: The case of communication networks and the power grid. *arXiv preprint arXiv:1304.0356*, 2013.

[16] Dung T Nguyen, Yilin Shen, and My T Thai. Detecting critical nodes in interdependent power networks for vulnerability assessment. 2013.

[17] Andrey Bernstein, Daniel Bienstock, David Hay, Meric Uzunoglu, and Gil Zussman. Power grid vulnerability to geographically correlated failures-analysis and control implications. *arXiv preprint arXiv:1206.1099*, 2012.

[18] Arunabha Sen, Anisha Mazumder, Joydeep Banerjee, Arun Das, and Randy Compton. Identification of k most vulnerable nodes in multi-layered network using a new model of interdependency. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 831–836. IEEE, 2014.

[19] Anamitra Pal, James S Thorp, Santosh S Veda, and VA Centeno. Applying a robust control technique to damp low frequency oscillations in the wecc. *International Journal of Electrical Power & Energy Systems*, 44(1):638–645, 2013.

[20] Tong Wang, Anamitra Pal, James S Thorp, Zengping Wang, Jizhen Liu, and Yuan Yang. Multi-polytope-based adaptive robust damping control in power systems using cart. *IEEE Transactions on Power Systems*, 30(4):2063–2072, 2015.

[21] Tong Wang, Anamitra Pal, James S Thorp, and Yuan Yang. Use of polytopic convexity in developing an adaptive interarea oscillation damping scheme. *IEEE Transactions on Power Systems*, 2016.

[22] Meiyan Li, Anamitra Pal, Arun G Phadke, and James S Thorp. Transient stability prediction based on apparent impedance trajectory recorded by pmus. *International Journal of Electrical Power & Energy Systems*, 54:498–504, 2014.

[23] Jean-Francois Castet and Joseph H Saleh. Interdependent multi-layer networks: Modeling and survivability analysis with applications to space-based networks. *PloS one*, 8(4):e60402, 2013.

[24] S Corsi and C Sabelli. General blackout in italy sunday september 28, 2003, h. 03: 28: 00. In *Power Engineering Society General Meeting, 2004. IEEE*, pages 1691–1702. IEEE, 2004.

[25] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Bull. Inst. Internat. Statist*, 38(4):343–347, 1961.

[26] Allen J Wood and Bruce F Wollenberg. *Power generation, operation, and control*. John Wiley & Sons, 2012.

[27] Pablo Echenique, Jesús Gómez-Gardeñes, and Yamir Moreno. Improved routing strategies for internet traffic delivery. *Physical Review E*, 70(5):056105, 2004.

[28] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Chapter iv network flows. *Handbooks in operations research and management science*, 1:211–369, 1989.

[29] Joseph Homer Saleh and Jean-François Castet. *Spacecraft reliability and multi-state failures: a statistical approach*. John Wiley & Sons, 2011.

[30] Jean-Francois Castet and Joseph H Saleh. Satellite reliability: statistical data analysis and modeling. *Journal of Spacecraft and Rockets*, 46(5):1065–1076, 2009.

[31] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on power systems*, 26(1):12–19, 2011.

[32] MT Hajiaghayi, K Jain, K Konwar, LC Lau, II Mandoiu, A Russell, A Shvartsman, and VV Vazirani. The minimum k-colored subgraph problem in haplotyping and dna primer selection. In *Proceedings of the International Workshop on Bioinformatics Research and Applications (IWBRA)*. Citeseer, 2006.

[33] Ettore Bompard, Enrico Pons, and Di Wu. Analysis of the structural vulnerability of the interconnected power grid of continental europe with the integrated power system and unified power system based on extended topological approach. *International Transactions on Electrical Energy Systems*, 23(5):620–637, 2013.