Reasoning Under Uncertainty in Cyber-Physical Systems: Toward Efficient and Secure Operation

By
Erik Miehling

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in the University of Michigan
2018

Doctoral committee:

Professor Demosthenis Teneketzis, Chair
Associate Professor Saurabh Amin, Massachusetts Institute of Technology
Professor George Cybenko, Dartmouth College
Assistant Professor Johanna Mathieu
Professor Michael Wellman

The purpose of models is not to fit the data but to sharpen the questions.

Samuel Karlin, 1924—2007

Erik Miehling

miehling@umich.edu

ORCID iD: 0000-0003-0533-8329

To my family.

Acknowledgments

There are many people to thank in my journey to obtain a doctorate. First of all, I would like to thank my advisor, Demosthenis Teneketzis. Thank you for your unending patience and for giving me time to discover my abilities. Your commitment to perform rigorous research has instilled values in me that will last my entire career. I can confidently say that I am now able to effectively filter out the signal from the noise. Barbara, I am very happy to have gotten to know you over the years. I truly enjoy the time that I have gotten to spend with you and Demos.

I would like to thank my committee: Saurabh Amin, George Cybenko, Johanna Mathieu, and Michael Wellman. Saurabh, your encouragement early on in my studies helped me to define and pursue my own research directions. George, your extensive wisdom across a wide variety of fields has always left me inspired after our conversations. Johanna, your patience in teaching me power systems is greatly appreciated; I would not possess the knowledge I do if it weren't for our interactions. Michael, your academic rigor and attention to detail has always been something I have admired. I am glad that our paths crossed during my time at Michigan.

I would also like to extend my gratitude to my colleagues. First, members of the group: Ouyang Yi, Hamidreza Tavafoghi, and Mohammad Rasouli. I am glad to have shared the journey through graduate school with you all. More recently, Farzaneh Farhadi, I am grateful to have gotten to know you over the past year and a half – I only wish that I had met you sooner. Members of the department, both past and present: Becky Turanski, Beth Lawson, Shelly Feldkamp, Judi Jones, and José-Antonio Rubio. You all worked tirelessly to ensure that us (absent-minded) students stayed on track. Members of the DCO: Joel VanLaven, Linda Randolph, Liz Zaenger, and Don Winsor; thank you for being patient with (sometimes trivial) technical issues and for providing me with access to the computational resources I needed to carry out my research. Joel, thanks for the entertaining and informative conversations.

Most importantly, I would like to thank my friends and family. Thank you to the friends I have met along the way: Seth K., Parinaz N., Dave R., Alex B., Rohit G., Deepanshu V., Roy D., and Katie D.C. To my friends back in Vancouver: Brandon W., Joe P., and Chris F., thank you for being great friends over the years while I've been away studying. Even with being away for months on end, we have always managed to pick up our friendship right where we left off. Thank you to my parents and sister for being so understanding and giving me time to focus on my research. I am looking forward to the day that I live closer and we can spend more time together. Last, but certainly not least, I would like to especially thank my own personal cheerleader, Monique M., for your understanding and support during the last few years of my degree. Your presence was essential in ensuring that I remained in a good state of mind. I couldn't have done it without you.

Table of Contents

List of Tables

List of Figures

List of Appendices

Abstract

The increased sensing, processing, communication, and control capabilities introduced by cyber-physical systems bring many potential improvements to the operation of society's systems, but also introduce questions as to how one can ensure their efficient and secure operation. This dissertation investigates three questions related to decision-making under uncertainty in cyber-physical systems settings.

First, in the context of power systems and electricity markets, how can one design algorithms that guide self-interested agents to a socially optimal and physically feasible outcome, subject to the fact that agents only possess localized information of the system and can only react to local signals? The proposed algorithms, investigated in the context of two distinct models, are iterative in nature and involve the exchange of messages between agents. The first model consists of a network of interconnected power systems controlled by a collection of system operators. Each system operator possesses knowledge of its own localized region and aims to prescribe the cost minimizing set of net injections for its buses. By using relative voltage angles as messages, system operators iteratively communicate to reach a social-cost minimizing and physically feasible set of injections for the whole network. The second model consists of a market operator and market participants (distribution, generation, and transmission companies). Using locational marginal pricing, the market operator is able to guide the market participants to a competitive equilibrium, which, under an assumption on the positivity of prices, is shown to be a globally optimal solution to the non-convex social-welfare maximization problem. Common to both algorithms is the use of a quadratic power flow approximation that preserves important non-linearities (power losses) while maintaining desirable mathematical properties that permit convergence under natural conditions.

Second, when a system is under attack from a malicious agent, what models are appropriate for performing real-time and scalable threat assessment and response selection when we only have partial information about the attacker's intent and capabilities? The proposed model, termed the dynamic security model, is based on a type of attack graph, termed a condition dependency graph, and describes how an attacker can infiltrate a cyber network. By embedding a state space on the graph, the model is able to quantify the attacker's progression. Consideration of multiple attacker types, corresponding to attack strategies, allows one to model the defender's uncertainty of the attacker's true strategy/intent. Using noisy security alerts, the defender maintains a belief over both the capabilities/progression of the attacker (via a security state) and its strategy (attacker type). An online, tree-based search method, termed the online defense algorithm, is developed that takes advantage of the model's structure, permitting scalable computation of defense policies.

Finally, in partially observable sequential decision-making environments, specifically partially observable Markov decision processes (POMDPs), under what conditions do optimal policies possess desirable structure? Motivated by the dynamic security model, we investigate settings where the underlying state space is partially ordered (*i.e.* settings where one cannot always say whether one state is better or worse than another state). The contribution lies in the derivation of natural conditions on the problem's parameters such that optimal policies are monotone in the belief for a class of two-action POMDPs. The extension to the partially ordered setting requires defining a new stochastic order, termed the generalized monotone likelihood ratio, and a corresponding class of order-preserving matrices, termed generalized totally positive of order 2.

Chapter 1

Introduction

Technology is increasingly finding its way into all aspects of our lives. Beyond our smartphones and computers, a growing number of devices and systems that we interact with on daily basis are intelligent, capable of gathering information from the real-world and processing it on-board in order to make real-time decisions and generate feedback. Examples range from personal voice assistants (*e.g.* Amazon's Alexa) and intelligent wearables (*e.g.* the Apple Watch) to the larger scale settings of autonomous vehicles and smart building management systems.

The feature of combining information processing with a real-world, physical system is a representative characteristic of a class of systems termed cyber-physical systems. Specifically, a cyber-physical system is one in which a physical system or process is "monitored, coordinated, controlled, and integrated" by a densely connected computation and communication network [Rajkumar et al., 2010]. Cyber-physical systems integrate sensing, information processing, communication, and control capabilities into all levels of the physical infrastructure with the aim of collecting a vast amount of information of the underlying system in order to realize large gains in operational efficiency. This integration has been made more feasible in recent years due to the shrinking size and cost of sensors and processors, a societal shift described by popular terms such as the internet-of-things and the internet-of-everything.

Society is increasingly recognizing the utility of cyber-physical systems for the design and efficient operation of critical infrastructure systems. These systems form an integral part of our modern lives, including the power systems that generate and distribute our electricity, the transportation networks that enable us to quickly and safely reach our destinations, the distribution networks that supply clean water to our homes, and the cellular and wireless networks that we all rely upon to remain connected. It is undeniable that leveraging the full capabilities of cyber-physical systems in these domains have the potential to drastically improve their efficiency, functionality, and profitability, resulting in beneficial effects on our economy and society as a whole. Cyber-physical systems are also expected to bring improvements to manufacturing, industrial control, factory automation, aerospace, and defense systems [Khaitan & McCalley, 2015].

The promised benefits of cyber-physical systems do not come without the introduction of some significant challenges and risks. The inherently distributed sensing, communication, and control capabilities of cyber-physical systems raise questions as to how one will be able to take full advantage of this new-found functionality, especially in the time-critical and large-scale domains present in many real-world applications. Furthermore, while the dense connectivity innate to cyber-physical systems enables devices to efficiently communicate information, it also opens up the possibility of malicious agents being able to exploit this functionality to their advantage and gain access to the system. As these systems grow and more physical components become instrumented with processing and communication capabilities, the set of attack pathways that a malicious agent can use to infiltrate the system (*i.e.* the attack surface) also unavoidably grows.

Addressing the above concerns involves developing algorithms, models, and theory that study and exploit the conjunction between processing and communication capabilities and the physical system. The analysis of cyber-physical systems uses ideas from decentralized and distributed optimization to address the distributed nature of information and computation, graph theory to describe the interconnections between system components and dependencies between operating conditions, and (stochastic) control theory and game theory to capture the uncertain effects of actions on the dynamics of the underlying system and the feedback of information to the decision-making process.

There are many rich and complex research questions that arise out of cyber-physical systems settings. This dissertation focuses on the informational aspect of these problems, specifically, how decisions are made when there is some uncertainty of the underlying system. This involves analyzing the structure of how information exists in the system and properties of how it is revealed to decision-makers in order to design models and algorithms that can efficiently translate all of the available information into decisions, while keeping an eye on tractability in realistic domains.

1.1. Decision-Making in Cyber Physical Systems under Imperfect Information

Inherent to problems in cyber-physical systems settings is the requirement to make decisions without necessarily having certainty of the current operating status or the underlying structure of the system. Often, such decisions must be made under the restriction that the information necessary for making the optimal decision does not reside in a single location or with a single agent, that is, there is no centralized, all-knowing entity. Instead, the information is distributed among multiple, potentially self-interested, decision-makers. Furthermore, due to the physical laws dictating the operation of the system and its inherently interconnected nature, the decisions of each agent have external, sometimes wide-spread, effects.

The presence of malicious agents adds another layer of uncertainty to the decision-making process. Malicious agents (termed attackers) have goals of their own, such as gaining access to sensitive information, commandeering key system components, or, more subtly, interfering with the agents' abilities to gather information (*e.g.* by corrupting existing data or injecting false data), impeding agents' abilities to perform accurate estimation and inference. Design of secure systems must go beyond the standard concerns of robustness to disturbances and random failures, but must also be capable of reasoning about attackers' abilities to maliciously interfere with the intended operation of the system and be able to take actions to ensure secure operation in the presence of such behavior.

This dissertation investigates three general questions related to decision-making under imperfect information in cyber-physical systems settings:

1) In the context of power systems and electricity markets, how can one design algorithms that guide self-interested agents to a socially optimal and physically feasible outcome, subject to the fact that agents only possess localized information of the system and can only react to local signals?

2) When a system is under attack from a malicious agent, what models are appropriate for performing real-time and scalable threat assessment and response selection when we only have partial information about the attacker's intent/strategy and capabilities?

3) In partially observable sequential decision-making environments, specifically partially observable Markov decision processes (POMDPs), under what conditions do optimal policies (functions mapping the decision-maker's belief of the system to an action) possess desirable structure?

Motivation, as well as the specific context, for each of these questions is described in more detail in the following section. The research contributions are also made explicit.

1.2. Problem Settings and Contributions

The problems studied in this dissertation focus on the development and analysis of models for decision-making under uncertainty subject to constraints arising from both physical considerations and the problem's information structure. This dissertation investigates this general theme in two main application areas: decentralized decision-making in the context of power systems and electricity markets, and sequential decision-making under uncertainty in the context of cyber-physical systems security. A central theme of my work is investigating how the information structure of the problem can be used to design efficient algorithms and gain insight into the form of optimal solutions. Along these lines, I have also investigated a more general question regarding the structure of optimal policies for POMDPs. The work involves providing conditions on the problem's parameters in order to ensure that the optimal policy has specific structure, shedding light on the relationship between the information pattern of the problem and the form of the optimal policy, as well as laying the groundwork for the design of efficient policy search algorithms.

1.2.1. Power Flow Algorithms and Electricity Market Mechanisms (Ch. 3 and 4)

Deregulation of the electric power industry has resulted in systems that consist of many self-interested agents. Under this setting, information is decentralized with each agent only possessing a localized view of the system. As a result, if agents were to make decisions in isolation, they would be unable to do so in a way that resulted in a feasible, let alone optimal, outcome for the system. To complicate matters, power flows through the network according to rules dictated by the laws of physics, creating system-wide coupling between variables and causing individual dispatch decisions to generate large and far-reaching externalities. Furthermore, the equations dictating power flow are highly nonlinear.

The contribution of Chapters 3 and 4 is in the development of provably convergent algorithms for obtaining socially optimal outcomes subject the aforementioned physical and informational constraints. Chapter 3 introduces a decentralized algorithm for determining the optimal net power injections at each bus (node in the network) in a multi-area power system. Each area is controlled by a system operator who is responsible for determining the set of net injections for its own region, subject to local feasibility conditions. Through an iterative message-exchange process (using relative voltage phase angles as messages) the system operators agree upon a set of power flows between adjacent regions that result in a socially optimal set of net injections. Chapter 4 introduces a more general model, consisting of many decision-makers, termed market participants – generation companies (GenCos), distribution companies (DistCos), and transmission companies (TransCos) – each with localized information of the system. Using the price of power at each bus as signals, the market mechanism involves the market participants reporting their surplus-maximizing outcomes, for a given set of prices, to a market operator, who is then responsible for updating the prices. The mechanism efficiently guides agents to an agreement, termed a *competitive equilibrium*, while respecting their informational constraints. Exploiting the structure of a quadratic approximation of the power flow equations, we are able to show that, under natural conditions (positivity of edge-wise price sums), the resulting competitive equilibrium is a global saddle-point of the Lagrangian and results in a globally optimal solution of the non-convex social welfare maximization problem.

1.2.2. Dynamic Security Strategies for Cyber-Physical Systems (Ch. 6)

Cyber-physical systems promise to greatly improve our quality of life, but will unavoidably come with the introduction of a myriad of vulnerabilities, allowing attackers to maliciously interfere with their intended operation. The scale of the attack surfaces in such systems, especially those of critical infrastructure, necessitates the development of automated defense systems that are capable of efficiently translating large amounts of noisy security alert information (from an intrusion detection system) into a quantification of the system's security status, with the goal of prescribing actions that prevent the attacker from achieving its goal(s). Models must be able to reason about all possible attack pathways that a malicious agent can use to infiltrate the system while permitting tractable computation of security strategies.

The contribution of this chapter is in the development of a partially observable sequential decision model for real-time threat assessment and response selection in cyber-physical systems. Sophisticated attacks unfold in a complex manner, involving the exploitation of vulnerabilities across multiple system components. In order to capture this behavior, the proposed model explicitly represents all attack pathways via a type of attack graph termed a condition dependency graph. The dependency graph allows one to reason about the current capabilities of the attacker and its proximity to its objectives. In the context of cyber-physical systems, the attacker's objectives represent conditions that permit the attacker to inflict damage to the physical infrastructure. Taking into account the cost of an attacker achieving its objective, as well as the cost of defense actions, one can cast the problem of determining optimal security strategies as a POMDP, where the information state (belief) is the joint distribution over the set of attacker's current capabilities and strategy. Scalability is achieved by employing an online, tree-based search method which involves simulating future possible scenarios, from the current history, in order to gain accurate estimates of the effectiveness of various defenses. Furthermore, taking advantage of the structure of observations, we are able to process a high volume of security alerts, enabling efficient inference in large-scale domains.

1.2.3. Structural Properties of Optimal Policies for POMDPs (Ch. 7)

POMDPs have enormous practical value. Unfortunately, solving them (*i.e.* obtaining an optimal policy) is typically a very computationally intensive task. Questions investigating conditions under which optimal policies have desirable structure are helpful for not only gaining insight into the optimal decision rule, but also allowing for the design of efficient policy search algorithms, pruning the space in which optimal policies live.

Our contribution lies in the derivation of natural conditions under which the optimal policy is *monotone* in the belief when the underlying state-space is partially ordered (*i.e.* motivated by the state space of the dynamic security model of Chapter 6, we investigate settings in which one cannot always say whether one state is better or worse than another state). Due to the partial ordering of the state-space, we propose a new stochastic order, generalizing the monotone likelihood ratio order. The stochastic order has many desirable properties, allowing us to establish monotonicity properties of the value functions and dynamic programming recursion, ensuring monotone optimal policies in a two-action setting. The work represents a contribution to the existing theory regarding structural properties of optimal policies for POMDPs.

1.3. Organization of the Dissertation

This work is divided into two parts. Each part begins with a preliminary chapter that provides necessary background information and gives context for the chapters that follow. In the first part, technical preliminaries for problems related to the electrical grid are provided in Chapter 2, followed by a decentralized algorithm for the operation of power systems in Chapter 3, and a decentralized mechanism for deregulated electricity markets in Chapter 4. The focus of these chapters is on determining social-cost minimizing (social welfare maximizing) outcomes under the condition that agents in the system only possess localized information and can only react to local signals. In the second part, Chapter 5 describes the issue of cyber-physical systems security in more detail, outlining the key features of these problems. Chapter 6 proposes a formal model for real-time threat assessment and response selection in large-scale cyber physical systems. Chapter 7 investigates a more general setting and derives conditions such that optimal policies in POMDPs are monotone in the belief. Closing remarks and views on future directions are provided in Chapter 8.

Part I

Decentralized Operation of Power Systems & Markets

Chapter 2

Power Systems & Markets

A power system is defined as a network of nodes $\mathcal{N}_b = \{1, \ldots, n_b\}$, termed buses in the power systems community, connected by transmission lines, denoted by the undirected edge-set $\mathcal{E}_l$. Each edge, $\{n, m\} \in \mathcal{E}_l$, has physical parameters described by a line limit $K_{nm} = K_{mn} > 0$ (capturing how much power flow it can sustain) and an admittance $Y_{nm} = G_{nm} + \mathbf{i}B_{nm}$ which consists of a conductance $G_{nm} = G_{mn} > 0$ and a susceptance $B_{nm} = B_{mn} > 0$. We set $K_{nm} = 0$ and $Y_{nm} = 0 + \mathbf{i}0$ for any $\{n, m\} \notin \mathcal{E}_l$ (*i.e.* any edge that doesn't exist in $\mathcal{E}_l$). Buses serve as a connection point for generators and loads to the rest of the network. Each bus can, in general, have both generators and loads connected to it.[*] The net injection at each bus is equal to difference between generation and demand at that bus, that is, the net injection at bus n is given by $I_n = p_n - s_n$, where p_n is the net generation at bus n and s_n is the net load (demand) at bus n. Each bus has two associated variables: a voltage magnitude, V_n, and a voltage phase angle, θ_n. The pair of voltage magnitudes and angles for all buses,

[*] A bus can also have no generators or loads, such a bus is termed a *zero-injection* bus.

written as $(\boldsymbol{V}, \boldsymbol{\theta})$, is termed the *operating point* of the system. Fig. 2.1 shows an example of a 3-bus power system.[†]

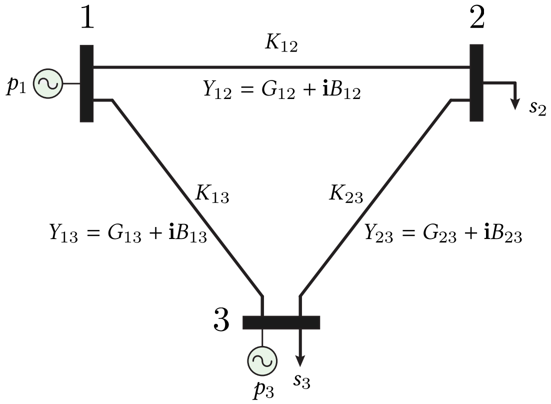

1
2
K_{12}
p_1
$Y_{12} = G_{12} + \mathbf{i}B_{12}$
s_2
K_{13}
K_{23}
$Y_{13} = G_{13} + \mathbf{i}B_{13}$
$Y_{23} = G_{23} + \mathbf{i}B_{23}$
3
p_3
s_3

Figure 2.1: A 3-bus power system example. Buses 1 and 3 have generators present, with generation levels given by p_1 and p_3, respectively. Buses 2 and 3 have loads, with demand levels of s_2 and s_3. Net injections at the buses are $I_1 = p_1$, $I_2 = -s_2$, and $I_3 = p_3 - s_3$. The operating point of the above system is given by $(\boldsymbol{V}, \boldsymbol{\theta}) = ((V_1, V_2, V_3), (\theta_1, \theta_2, \theta_3))$.

The amount of power flowing along a line is given by the AC power flow equations. Specifically, the (real) power flowing from bus n to bus m, denoted by P_{nm}, is given by (from [Elgerd, 1973]).[‡]

$$P_{nm} = G_{nm}V_n^2 - G_{nm}V_nV_m \cos(\theta_n - \theta_m) + B_{nm}V_nV_m \sin(\theta_n - \theta_m). \tag{2.1}$$

Due to the nonlinearity of the AC equations, it is common to use approximations. A well-known approximation, termed the DC approximation, sets all voltages to 1 (per unit, p.u.) and uses the small-angle approximations $\sin(\theta_n - \theta_m) \approx \theta_n - \theta_m$ and $\cos(\theta_n - \theta_m) \approx 1$, to obtain an approximate flow expression between two buses, n and m, as $P_{nm}^{DC} = B_{nm}(\theta_n - \theta_m)$.

[†] Power systems are usually drawn as a *single-line diagram*, as seen in Fig. 2.1.

[‡] Note that we only consider real power in our model.

While the DC approximation is simple and permits efficient computation, it does have some drawbacks (details in Appendix A.1). We make use of an alternative approximation, which we term the *modified DC approximation*, which is described in Section 2.2.

The net injection at each bus, computed as the difference between the net generation and demand at the bus, must agree with the injections due to the operating point $(\boldsymbol{V}, \boldsymbol{\theta})$. This requirement describes the physical laws of power flow, and is represented by the *power balance equation*. First, as stated above, the net injection at each bus due to generation and demand is given by $I_n = p_n - s_n$. Second, the operating point induces an injection at each bus n dictated (under the AC power flow equations) by the following equation.

$$f_n(\boldsymbol{V}, \boldsymbol{\theta}) = \sum_{m \in \mathcal{N}_b} G_{nm} V_n^2 - G_{nm} V_n V_m \cos(\theta_n - \theta_m) + B_{nm} V_n V_m \sin(\theta_n - \theta_m).$$

The power balance equation states that these two injections must agree at each node, and is thus given by

$$I_n = p_n - s_n = f_n(\boldsymbol{V}, \boldsymbol{\theta}). \tag{2.2}$$

The existence of the power balance equation makes power systems a difficult class of networks to analyze. The following section elaborates on some of these difficulties.

2.1. The Nature of Power Flow

The set of net power injections at the buses in the network correspond to a physical operating point, as dictated by the power balance equation, Eq. (2.2). Modification of the injection at a single bus will induce a corresponding change in the operating point of the entire system, in turn, requiring a modification of the injections at other buses in order to ensure that balance in the network is maintained. In centralized information settings, the (single) decision-maker knows the structure and parameters of the network and is able to completely capture these effects. That is, it is able to specify a set of injections and an operating point of the system, such that the combination is physically consistent.

One can see how this causes an issue when information is decentralized, that is, when each decision-maker only possesses knowledge of a localized region of the network. In this setting, if each decision-maker were to specify a set of injections for its region of the network, it would not be able to do so in a way that would be physically feasible for the system. We argue that, under the decentralized information setting, decision-makers should not propose power injections at buses, rather they should propose the operating point of their localized region of the network.[§] One can see, by inspecting the power balance equation, that the injection at bus n is completely characterized by the voltage magnitudes and angles at, and immediately neighboring (buses with a connected line), bus n. This way, each decision-maker is able to propose a set of voltage magnitudes and relative angles that is physically consistent with their localized region of the network. The algorithms proposed in the first part of this dissertation both take advantage of this idea.

2.2. Modified DC Approximation

We consider a power flow approximation, similar to that of Chao & Peck [Chao & Peck, 1996], that represents power flow between two buses as a convex function of the voltage angle difference. To begin the derivation, recall that, by the AC power flow equation, the real power flowing from bus n to bus m is $P_{nm} = G_{nm}V_n^2 - G_{nm}V_nV_m\cos(\theta_n - \theta_m) + B_{nm}V_nV_m\sin(\theta_n - \theta_m)$. We set voltage magnitudes to 1 p.u., $V_n = 1\ \forall n \in \mathcal{N}_b$, and assume that voltage angle differences, $\theta_n - \theta_m$, are small (similar to the DC approximation). However, unlike the DC approximation, we use *second-order* small angle approximations, $\sin(\theta_n - \theta_m) \approx \theta_n - \theta_m$ and

[§] Inherent to all decentralized decision-making problems is the need to iteratively communicate with other agents in order to reach an agreement. This communication process relies on the communication of *messages* or *proposals*.

$\cos(\theta_n - \theta_m) \approx 1 - \frac{1}{2}(\theta_n - \theta_m)^2$, writing the expression for the power flow from bus n to bus m as a convex function of the angle difference, $\theta_n - \theta_m$. The resulting approximation, which we term the modified DC approximation, dictates that the flow of power on line (n, m) is

$$g(\theta_{nm}) := B_{nm}(\theta_n - \theta_m) + \frac{1}{2}G_{nm}(\theta_n - \theta_m)^2 \tag{2.3}$$

where $\theta_{nm} = \theta_n - \theta_m$. This simple modification of the DC approximation maintains the asymmetry of the power flow equations, $g(\theta_{nm}) \neq -g(\theta_{mn})$, and consequently allows for power losses to be considered (unlike with the DC power flow approximation). The real power losses along line $\{n, m\}$, $L_{nm} = P_{nm} + P_{mn}$, are approximated by $L_{nm} \approx G_{nm}(\theta_n - \theta_m)^2$. For notational convenience, we split Eq. (2.3) into a DC component, $\bar{g}(\theta_{nm}) := B_{nm}(\theta_n - \theta_m)$, and a (convex) loss component, $\tilde{g}(\theta_{nm}) := \frac{1}{2}G_{nm}(\theta_n - \theta_m)^2$. The accuracy of the above approximation, Eq. (2.3), is demonstrated through load flow analyses on multiple test systems (results in Appendix A.1).

2.3. The Issue of Power Losses

The modified DC approximation allows for power losses to be approximately captured, offering improved accuracy over the loss approximations in the literature. The inclusion of power losses in optimal power flow problems is crucial for obtaining a realistic dispatch solution, especially in large and heavily-loaded networks. Furthermore, in the context of electricity markets, accurate modeling of losses is key for obtaining prices of power across the grid that are representative of the true operating point. Ideally, one would perform an optimal power flow analysis using the nonlinear (AC) power flow equations (in a centralized setting) in order to obtain the true power losses in the transmission network; however, in pursuit of simpler and more computationally-friendly methods, multiple attempts at estimating the line losses have been developed in the literature. The main approaches for estimating power losses involve: ***1)*** augmenting load with an a priori estimate of losses, ***2)*** representating total system losses as a quadratic function of the net power injection vector (through the B-coefficient loss expression [Kirchmayer, 1958, Wood & Wollenberg, 2012], also known as Kron's loss formula, and qualitatively similar approach in [Aoki & Satoh, 1982]), ***3)*** including penalty functions in the objective function [Fan & Zhang, 1998, Chen & Chen, 2003], and ***4)*** providing individual loss expressions for each line [Alguacil & Conejo, 2000, Motto et al., 2002b, dos Santos & Diniz, 2011, Wood & Wollenberg, 2012]. First, a priori estimation of losses is difficult due to physical laws and the nonlinear nature of power flows, making an accurate estimation of losses a futile task for large networks. The second approach, of representing losses as quadratic functions of the net injection vector, can produce reasonable approximations for total system losses; however, due to the fact that the coefficients in the quadratic expression (the B-coefficients in [Kirchmayer, 1958, Wood & Wollenberg, 2012]) are computed for a fixed operating point, the accuracy of the method can suffer significantly when the operating point changes. Third, penalty methods represent transmission losses as penalty terms in the cost functions of generators. These penalties are determined by computing an *incremental transmission loss* (ITL) coefficient, a process that can be difficult and somewhat arbitrary. Lastly, the most accurate of the aforementioned methods, is via individual loss expressions for each line. Individual loss expressions represent line losses as a function of the operating point directly. This property, while permitting a very accurate approximation, introduces some difficulties from an operational perspective. Multiple papers involve methods that address these difficulties. Alguacil and Conejo [Alguacil & Conejo, 2000] formulate a multiperiod optimal power flow problem which uses individual loss expressions. The loss functions are formulated using cosines leading to a nonlinear optimization problem which is solved via Bender's decomposition. Motto et al. [Motto et al., 2002b] form a second-order approximation to the cosine term in [Alguacil & Conejo, 2000], much like the modified DC approximation; however, motivated by computational reasons, they further approximate the quadratic expression by a piecewise linear function. A similar approach is taken in [dos Santos & Diniz, 2011] which offers improved accuracy over [Motto et al., 2002b] by choosing the linearized segments iteratively.

The approach taken in this dissertation of expressing line losses as a convex function of the angle difference offers advantages in both accuracy and computational efficiency over the methods in the literature. For instance, the loss approximation of our work does not suffer from inaccuracies when the operating point changes, unlike the B-coefficient method. Additionally, we can avoid the errors introduced via (piecewise) linearization of the loss expressions in [Motto et al., 2002b, dos Santos & Diniz, 2011]. Furthermore, our approach leads to a *convex problem*, as opposed to the computationally difficult nonlinear problem found in [Alguacil & Conejo, 2000].

Many of the approaches taken in the literature either deal with the fully nonlinear flow expression or attempt to obtain a linearized form. We argue that a reasonable middle ground, that of convexity, allows for one to make theoretical guarantees (convergence of algorithms) while still accurately describing important nonlinearities of the problem.

2.4. Overview of Part I

The remainder of Part I focuses on the development of models for the decentralized operation of power systems and electricity markets. The model of Chapter 3 describes an electrical grid with multiple control areas, each one containing multiple buses and transmission lines, with each area operated by a distinct decision-maker, termed a *system operator* (SO). In this system-of-systems setting, we study the problem of how to determine a cost-minimizing set of net power injections for the buses in each region, subject to the fact that each system operator only possesses localized information. Since the regions are connected, determining such injections requires each SO to determine the appropriate power trades with their adjacent SOs. Using the modified DC approximation, the net injection at each bus can be fully described using its own voltage angle and the angles of its neighboring buses. The resulting collection of optimization problems are convex and can efficiently be solved iteratively using a well-known distributed optimization algorithm (the *alternating direction method of multipliers* (ADMM)).

In Chapter 4, a mechanism for achieving an efficient outcome in deregulated electricity markets is developed. The model consists of a market operator (MO) and multiple, self-interested market participants. Each market participant possesses localized information of the system and can only react to local price signals. Based on the dual decomposition algorithm, we develop a provably convergent market mechanism that achieves a Pareto efficient outcome.

Chapter 3

A Decentralized Multi-Area Optimal Power Flow Algorithm with Power Losses

3.1. Introduction

In this chapter, we consider a network of interconnected power systems, run by *system operators* (SOs), where the goal is to determine a cost-minimizing set of injections subject to the constraint that each SO does not know the structure of the system outside of its own localized region. Due to the interconnected nature of the problem, determining these injections requires that SOs exchange power with adjacent SOs. The proposed algorithm, based on the *alternating direction method of multipliers* (see [Boyd et al., 2011]), dictates that SOs solve their respective localized-information problem and iteratively communicate the shared components of their solutions (voltage angles) with adjacent SOs, eventually converging to an agreed-upon set of voltage angles.* The convergent set of angles induce power trades between SOs and result in a socially-optimal set of net injections for the system.

3.1.1. Literature Review

Problems related to determining the lowest cost generation that satisfies demand, subject to the physical constraints of the system, are referred to in the power systems community as *optimal power flow* (OPF) problems. OPF problems under centralized information have been studied extensively since the problem's inception [Carpentier, 1962] and the resulting literature is vast. The literature review in this chapter includes the most relevant works; the interested reader is referred to [Pandya & Joshi, 2008, Frank et al., 2012a, Frank et al., 2012b] for more complete reviews. A popular method for obtaining the solution to the OPF problem under the AC power flow equations (the general AC-OPF problem) involves forming its semidefinite program (convex) relaxation [Bai et al., 2008, Lavaei & Low, 2012]. While an attractive approach (the relaxation allows for a polynomial time solution), it is known that when the duality gap is non-zero, which can occur in many practical examples, the resulting solution is not feasible [Molzahn et al., 2013]. Some additional methods for solving the OPF problem under centralized information include convex relaxation techniques [Low, 2013, Farivar & Low, 2013a, Farivar & Low, 2013b] and the holomorphic embedded load-flow method [Trias, 2012].

Solutions to the OPF problem under decentralized information have been investigated in many papers utilizing a wide variety of solution techniques. Many of the approaches use techniques from distributed optimization to decompose the global optimization problem into separate components which are then solved iteratively. Examples include approaches using dual decomposition methods [Baldick et al., 1992, Conejo & Aguado, 1998, Galiana

*It is important to note that *we are not proposing direct control of the voltage angles of the buses*; rather, we are using the voltage angles merely as the messages for each SO. Once the SOs reach an agreement, the convergent angles uniquely specify the lossy net injection (controllable variable) at each bus.

et al., 2002, Motto et al., 2002a, Biskas & Bakirtzis, 2004], augmented Lagrangian methods [Batut & Renaud, 1992, Kim & Baldick, 1997, Baldick et al., 1999, Kim et al., 2001, Bakirtzis & Biskas, 2002], and approximate Newton directions [Conejo et al., 2002, Nogales et al., 2003, Biskas et al., 2005, Hug-Glanzmann & Andersson, 2009]. Recently, the ADMM algorithm, a method combining the decomposability properties of the dual decomposition method and the robustness of augmented Lagrangian methods [Boyd et al., 2011], has seen much attention in the power systems community. Applications of the ADMM algorithm to problems of decentralized information in power systems settings include: multi-area unit commitment [Chung et al., 2011], decentralized optimal power flow for mesh networks [Kim & Baldick, 2000, Sun et al., 2013, Kraning et al., 2013, Dall'Anese et al., 2013, Mosca, 2013, Erseghe, 2014, Magnusson et al., 2014] and radial networks [Dall'Anese et al., 2013, Šulc et al., 2014, Peng & Low, 2014, Christakou et al., 2015], and distributed power system state estimation [Kekatos & Giannakis, 2013].

Convergence of decentralized OPF algorithms is a primary concern. The aforementioned algorithms are known to converge under convexity; however, many of the papers consider settings that are inherently non-convex (*i.e.* the general OPF problem), resulting in authors demonstrating convergence on a small number of test systems [Batut & Renaud, 1992, Kim & Baldick, 1997, Conejo & Aguado, 1998, Baldick et al., 1999, Kim & Baldick, 2000, Kim et al., 2001, Galiana et al., 2002, Motto et al., 2002a]. Some papers investigate sufficient conditions for convergence, but cannot guarantee convergence to a globally optimal solution [Baldick et al., 1992, Nogales et al., 2003, Hug-Glanzmann & Andersson, 2009]. Other papers attempt to apply the ADMM algorithm to the general AC-OPF problem, but can only guarantee convergence when the duality gap is zero [Erseghe, 2014] or convergence to local optima [Sun et al., 2013, Magnusson et al., 2014]. Other approaches employ the DC approximation and can consequently guarantee convergence [Bakirtzis & Biskas, 2002, Bakirtzis & Biskas, 2003, Biskas & Bakirtzis, 2004, Biskas et al., 2005, Mosca, 2013]; however, the convergence guarantee comes at the cost of ignoring important non-linearities in the model, for example, not being able to consider power losses. The approach by [Kraning et al., 2013] involves taking a convex hull of the non-convex constraints, permitting convergence, but resulting in the solution potentially not being feasible. Desirable convergence properties are obtained in [Šulc et al., 2014, Peng & Low, 2014, Christakou et al., 2015] but require one to restrict attention to radial networks. The ADMM algorithm has also been investigated in combination with the semidefinite relaxation approach [Dall'Anese et al., 2013], but as with the centralized information approach, a zero duality gap solution is required to ensure feasibility.

3.1.2. Contribution

As discussed above, existing approaches either deal with the full complexity of the AC-OPF problem, precluding convergence guarantees, or consider simplified settings, such as the DC approximation or restricted network topologies, limiting their accuracy and applicability. The approach taken in this chapter offers a simple convex approximation of the OPF problem that preserves some important non-linearities of the problem (such as power losses) while permitting convergence in general (mesh) networks.

3.2. The Multi-Area Power System Model

Throughout the discussion of the model the reader is directed to Fig. 3.1, in Section 3.5, which represents an instance of an interconnected power system topology. We consider a network of $n_{so} \geq 2$ *system operators* (SOs), denoted by the set $\mathcal{M}$. Each SO$_a$, $a \in \mathcal{M}$, contains a set of $n_b^a \geq 1$ unique buses, denoted by the set $\mathcal{N}_b^a$. The buses in the network are numbered sequentially based upon their SO index, that is, $\mathcal{N}_b^1 := \{1, \dots, n_b^1\}$, $\mathcal{N}_b^2 := \{n_b^1+1, \dots, n_b^1+n_b^2\}$, and so on, up to $\mathcal{N}_b^{n_{so}} := \{n_b^1 + \dots + n_b^{n_{so}-1} + 1, \dots, n_b\}$, where $n_b := n_b^1 + \dots + n_b^{n_{so}}$ is the total number of buses in the system. The set of all buses is denoted by the set $\mathcal{N}_b$. Due to the physical locations of loads and generators, some buses are a priori specified as net consumption or net generation buses. Net *consumption buses* (buses that contain only loads) are denoted by $\mathcal{N}_b^d$, whereas net *generation buses* (buses that contain only generators) are denoted by $\mathcal{N}_b^g$. All remaining buses are assumed to contain at least one load and at least one generator[†], which we term *hybrid buses*, and belong to the set $\mathcal{N}_b \setminus (\mathcal{N}_b^d \cup \mathcal{N}_b^g)$. We assume a set of slack buses (at most one per system, further discussed in assumption 2 in Section 3.2.1), denoted by $\mathcal{N}_b^s$, which serve only as angle reference buses. We assume that each slack bus has a generator present, that is $\mathcal{N}_b^s \subseteq \mathcal{N}_b \setminus \mathcal{N}_b^d$. For notational convenience, we introduce the following terminology for edges in the network. We term the network connecting buses of each SO_a as the *intra-SO_a* network with undirected set of edges $\mathcal{E}_l^a$. We term the network between buses that connect two SOs, for example SO_{a_1} and SO_{a_2}, as the *inter-SO_{a_1,a_2}* network, where the set of undirected edges between SO_{a_1} and SO_{a_2} are denoted by $\mathcal{E}_l^{a_1,a_2}$. Lines in the inter-SO_{a_1,a_2} network are also referred to as *tie-lines*.

We define the set of neighboring buses to bus n by $\mathcal{R}_n$, with $n \in \mathcal{R}_n$. We denote by $\bar{\mathcal{R}}_n$ as the set $\mathcal{R}_n$ with index n removed. We denote the set of buses in and immediately connected to buses in SO_a as $\mathcal{R}^a := \bigcup_{n \in \mathcal{N}_b^a} \mathcal{R}_n$. We define the set of adjacent SOs to SO_a as $\mathcal{M}^a$, that is, $\mathcal{M}^a$ is the set of SOs that contain at least one bus that is connected to a bus in $\mathcal{N}_b^a$, with $a \notin \mathcal{M}^a$. We associate a voltage angle with each bus n, denoted by θ_n. The vector $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^{n_b^1 + \cdots + n_b^{n_{so}}}$ is the complete set of bus angles across the network where Θ is the feasible set of angles. We use $\boldsymbol{\theta}_{\mathcal{R}_n}$ to represent the set of angles connected to (and including) bus i, not including any slack indices (since these angles are fixed), that is $\boldsymbol{\theta}_{\mathcal{R}_n} := \{\theta_m : m \in \mathcal{R}_n \setminus \mathcal{N}_b^s\}$. We also define $\boldsymbol{\theta}_{\mathcal{R}^a}$ as the vector of angles of buses in and immediately connected to SO_a, that is $\boldsymbol{\theta}_{\mathcal{R}^a} := \{\theta_m : m \in \mathcal{R}_n \setminus \mathcal{N}_b^s, n \in \mathcal{N}_b^a\}$, again not including any slack indices. Notice that there is coupling between variables of two adjacent SOs a and b, that is, $\boldsymbol{\theta}_{\mathcal{R}^a}$ and $\boldsymbol{\theta}_{\mathcal{R}^b}$ share some common variables from $\boldsymbol{\theta}$. As a result, we

[†] Note that topologies that contain zero injection buses are not permitted in the model of this chapter (discussed in more detail in assumption 1 in Section 3.2.1).

distinguish between each SO's copy of the shared variables by denoting elements of $\boldsymbol{\theta}_{\mathcal{R}^a}$ that are shared with another SO by $\theta_n^{(a)}$ and elements that are not shared simply by θ_n. The reader is referred to the caption of Fig. 3.1 for an example of the SOs' decision variables.

3.2.1. Model Assumptions

We assume that the power flow obeys the modified DC power flow approximation defined in Section 2.2. Additionally, we make the following three assumptions for this chapter's model.

Assumption 1 *(controllability of net injections)*: The net power injection I_n at each bus n is assumed to be controllable within a bus-specific range, $[p_n^{\min}, p_n^{\max}]$. The bounds $p_n^{\min}$ and $p_n^{\max}$ are defined by the feasible ranges of each generator and load. The feasible range of each generator is defined as its minimum operating generation output to its maximum generation capacity. With respect to loads, we assume that there are both fixed loads and flexible loads (the level of demand can be adjusted within some range; this is reasonable with the advent of widespread demand response capabilities) in the network. Implicit to assumption 1 is that each bus contains either a generator or a flexible load (or both). Consequently, zero-injection buses and buses with only fixed loads are not permitted in the model of this chapter (the model of Ch. 4 removes this requirement). The ranges of generators and flexible loads translate into controllability ranges on net injections at each bus. These constraints take the form

$$f_n(\boldsymbol{\theta}_{\mathcal{R}_n}) = \sum_{m \in \mathcal{N}_b} g(\theta_{nm}) \leq p_n^{\max}, \tag{3.1}$$

$$\bar{f}_n(\boldsymbol{\theta}_{\mathcal{R}_n}) = \sum_{m \in \mathcal{N}_b} \bar{g}(\theta_{nm}) \geq p_n^{\min} \tag{3.2}$$

where f_n represents the net power injections, and $\bar{f}_n$ represents the non-lossy net power injections at bus n (recall the discussion following Eq. (2.3) for the definition of $\bar{g}(\theta_{nm})$). The upper bound is placed on the net injection which ensures that the (upper) production bounds of the generators are satisfied. The lower bound is placed on the non-lossy injection in order to maintain convexity of problem.[‡]

Assumption 2 *(slack buses)*: Every SO either contains or is immediately connected to exactly one slack (reference) bus. That is, $\mathcal{R}^a \cap \mathcal{N}_b^s$ contains exactly one element for each $a \in \mathcal{M}$. These buses are termed *area slack buses*. Each SO is assumed to know the location of its area slack bus. The slack buses serve as reference buses with each voltage angle fixed to a reference value of zero, $\theta_n = 0$ for all $n \in \mathcal{N}_b^s$. Implicit to this assumption is that SOs agree upon the same reference value *a priori* and keep this reference fixed for the duration of the problem.

Assumption 3 *(cost functions)*: Each bus n has an associated cost function $c_n : \mathbb{R} \to \mathbb{R}$ which is assumed to be twice continuously differentiable[§], convex, and strictly increasing. The cost function of a bus, denoted by c_n, is the sum of the cost functions of generators and the (negative) benefit functions of loads at the bus. The interpretation of the each cost function is the same as the one used in [Wu & Varaiya, 1999]; if bus n's net injection I_n is positive, then $c_n(I_n)$ represents the *generation cost* of producing power I_n, whereas, if bus n's net injection I_n is negative, then $c_n(I_n)$ represents the *negative of the benefit* from receiving power I_n. See [Stott et al., 1987] for a discussion of the validity of the convexity assumption.

3.2.2. Knowledge Model

We now describe the *knowledge model*, that is, what each of the power system entities knows about the system. Each SO possesses localized knowledge of their own system (control

[‡] Ideally, these constraints should take the form $f_n(\boldsymbol{\theta}_{\mathcal{R}_n}) \geq p_n^{\min}$; however, since f_n is a convex function, constraints of this form generate non-convex sets. It is important to note that, since real power losses are always positive, constraint (3.2) implies $f_n(\boldsymbol{\theta}_{\mathcal{R}_n}) \geq p_n^{\min}$.

[§] For simplicity; the results still hold if the c_n's are not smooth.

area). Specifically, each SO_a possesses private information regarding the cost functions c_n and injection bounds $p_n^{\min}, p_n^{\max}$ for all buses in their system $n \in \mathcal{N}_b^a$. Each SO_a also knows the admittances Y_{nm}, line limits K_{nm}, and stability bounds $\underline{\theta}_{nm}$, $\overline{\theta}_{nm}$, of its localized region $\{n, m\} \in \mathcal{E}_l^a \cup \bigcup_{b \in \mathcal{M}^a} \mathcal{E}_l^{ab}$ (note that this includes information regarding the tie-lines). Additionally, SO_a knows the location of its area slack bus (as described in assumption 2 in Section 3.2.1).

3.3. The Multi-Area Optimal Power Flow Problem

The goal of the *multi-area optimal power flow problem* is to determine the net injections that induce the optimal tie-line flows among the interconnected power systems while satisfying the physical and informational constraints. The optimal tie-line flows are defined as the flows that are induced by the social-cost-minimizing set of injections. First, in Section 3.3.1, we formulate the centralized information problem, termed Problem (P_C), where we assume that there is an entity that has complete system knowledge. The solution to the centralized information problem defines the optimal social cost. Second, in Section 3.3.2, we consider an alternate formulation of Problem (P_C), termed the decentralized information problem, Problem (P_D), by introducing both local variables and a common global variable. Later, in Section 6.3, we present a message exchange process that results in the optimal cost (the solution of the centralized information problem) while obeying the assumptions of the knowledge model (see Section 3.2.2).

3.3.1. Centralized Information Problem Formulation

We now formulate the centralized information problem assuming that there is an entity that has complete knowledge of the network topology and system parameters. Recall that every bus has an associated cost function; a cost for generating power, or a negative benefit for receiving power. The *social cost* is defined to be the sum of all buses' costs across the system. As mentioned earlier, the optimal tie-line flows are those that are induced by the net injections that achieve the minimum social cost.

The centralized information problem (P_C) aims to determine the set of net power injections, $\boldsymbol{I} = (I_1, \dots, I_{n_b})$, such that the total social cost is minimized subject to the physical constraints.

$$
\begin{aligned}
\underset{\boldsymbol{I}=\{I_n\}_{n\in\mathcal{N}_b}}{\text{minimize}} \quad & \sum_{n\in\mathcal{N}_b} c_n(I_n) && (P_C)\\
\text{subject to} \quad & \boldsymbol{I} = \boldsymbol{f}(\boldsymbol{\theta}) && (P_C\text{-1})\\
& \boldsymbol{\theta} \in \Theta && (P_C\text{-2})
\end{aligned}
$$

where $\boldsymbol{I}$ represents the net injection vector and $\boldsymbol{f}(\boldsymbol{\theta}) = (f_1(\boldsymbol{\theta}_{\mathcal{R}_1}), \dots, f_{n_b}(\boldsymbol{\theta}_{\mathcal{R}_{n_b}}))$ represents the injection induced by the operating point $\boldsymbol{\theta}$. Let us denote the optimal solution of Problem (P_C) by $\boldsymbol{I}^*_C$ with corresponding objective value c^*_C. Under the modified DC approximation, the operating point is defined as the set of voltage angles, denoted by $\boldsymbol{\theta}$. Constraint (P_C-1) thus represents the power balance equation under the modified DC approximation. Constraint (P_C-2), $\boldsymbol{\theta} \in \Theta$, imposes the physical constraints of the system, and is defined as

$$
\Theta := \left\{(\boldsymbol{\theta}_{\mathcal{R}^1}, \dots, \boldsymbol{\theta}_{\mathcal{R}^{n_{so}}}) \in \Theta_{\mathcal{R}^1} \times \cdots \times \Theta_{\mathcal{R}^{n_{so}}} : \theta^{(a)}_n = \theta^{(a')}_n, i \in \mathcal{R}^a \cap \mathcal{R}^{a'}, a, a' \in \mathcal{M}\right\}
$$

where each SO$_a$'s feasible set, $\boldsymbol{\Theta}_{\mathcal{R}^a}$, is

$$
\begin{aligned}
\boldsymbol{\Theta}_{\mathcal{R}^a} := \Big\{\boldsymbol{\theta}_{\mathcal{R}^a} \Big| \; & \theta_n \in [-\pi, \pi], n \in \mathcal{R}^a \setminus \mathcal{N}_b^s; && (3.3)\\
& \theta_n = 0, n = \mathcal{R}^a \cap \mathcal{N}_b^s; && (3.4)\\
& f_n(\boldsymbol{\theta}_{\mathcal{R}_n}) \leq p_n^{\max}, \; n \in \mathcal{N}_b^a; && (3.5)\\
& \bar{f}_n(\boldsymbol{\theta}_{\mathcal{R}_n}) \geq p_n^{\min}, \; n \in \mathcal{N}_b^a; && (3.6)\\
& \underline{\theta}_{nm} \leq \theta_{nm} \leq \overline{\theta}_{nm}, m \in \mathcal{R}_n, n \in \mathcal{N}_b^a; && (3.7)\\
& g(\theta_{nm}) \leq K_{nm}, g(\theta_{mn}) \leq K_{mn}, m \in \mathcal{R}_n, n \in \mathcal{N}_b^a \Big\}. && (3.8)
\end{aligned}
$$

The first set of constraints (3.3-3.4) are the linear *voltage angle constraints*; trivial bounds are placed on all non-slack bus angles, with indices $\mathcal{R}^a \setminus \mathcal{N}_b^s$, whereas the angle of SO$_a$'s area slack bus is fixed to zero. The *maximum injection constraints*, (3.5), place an upper bound on lossy net injections, and *minimum injection constraints*, (3.6), place a lower bound on non-lossy net injections at each bus, as discussed in assumption 1. *Voltage angle stability constraints*, (3.7), $\underline{\theta}_{nm} \leq \theta_{nm} \leq \overline{\theta}_{nm}$ for all $\{n, m\} \in \mathcal{E}_l$, are in place to maintain synchronism throughout the system. Quantities $\underline{\theta}_{nm}, \overline{\theta}_{nm}$ are the maximum allowable angle differences in order to maintain stability of the system. The maximum theoretical stability bounds, $\underline{\theta}_{nm}, \overline{\theta}_{nm}$, are $\pm\pi/2$ radians (for lossless lines); however, the precise stability bounds depend upon installed equipment, its configuration, as well as transient stability considerations throughout the network [Cain et al., 2012]. The remaining set of constraints, (3.8), termed *line limit constraints*, specify that power flow must be within the limits of each line. Since $\boldsymbol{\Theta}_{\mathcal{R}^a} \subseteq \mathbb{R}^{|\mathcal{R}^a|-1}$ consists of linear equalities, linear inequalities, and convex inequalities, it is a convex and compact set, implying that $\boldsymbol{\Theta}$ is compact. We assume that $\boldsymbol{\Theta}$ is non-empty.

Problem (P_C) can be transformed into an equivalent problem that is expressed solely in terms of voltage angles. This is done by moving constraint (P_C-1) into the objective function, resulting in the optimization problem (P'_C),

$$\underset{\boldsymbol{\theta}\in\Theta}{\text{minimize}} \qquad \sum_{a\in\mathcal{M}} C_a(\boldsymbol{\theta}_{\mathcal{R}^a}) \tag{P'_C}$$

The objective function, $C(\boldsymbol{\theta}) = \sum_{a\in\mathcal{M}} C_a(\boldsymbol{\theta}_{\mathcal{R}^a})$, termed the *social cost function*, is expressed in terms of functions $C_a : \mathbb{R}^{|\mathcal{R}^a|-1} \to \mathbb{R}$, where each C_a is termed SO$_a$'s *aggregated cost function*, given by

$$C_a(\boldsymbol{\theta}_{\mathcal{R}^a}) = \sum_{n\in\mathcal{N}_b^a} c_n\left(f_n(\boldsymbol{\theta}_{\mathcal{R}_n})\right). \tag{3.9}$$

The following lemma regarding the convexity of the each SO's aggregated cost function will be useful in later demonstrating convergence properties of the proposed algorithm.

Lemma 3.3.1. *Each SO$_a$'s aggregated cost function, $C_a(\boldsymbol{\theta}_{\mathcal{R}^a})$, is strongly convex on $\Theta_{\mathcal{R}^a}$, $a \in \mathcal{M}$.*

Proof: See Appendix A.2.

Consequently, the social cost function $C(\boldsymbol{\theta})$ is also strongly convex in $\boldsymbol{\theta}$. The solution of Problem (P'_C) uniquely defines the optimal net power injections (control variables) for each bus and thus solves Problem (P_C).

3.3.2. Decentralized Information Problem Formulation

We wish to determine the feasible net injections that minimize the social cost under the informational constraints imposed by the problem structure (see Section 3.2.2). As in the discussion of the centralized problem, we can express the problem of finding the optimal net injections (those that minimize $\sum_n c_n(p_n)$ subject to constraints (P_C-1) and (P_C-2)) as a problem of finding the voltage angles that induce the injections. Instead of having one decision variable, the variable $\boldsymbol{\theta}$ in Problem (P'_C), we introduce *local variables* for each SO$_a$,

$a \in \mathcal{M}$, denoted by $\boldsymbol{\theta}_{\mathcal{R}^a}$, and a *global variable* $\mathbf{z}$. This modified problem, which we term the decentralized information problem, denoted by Problem (P_D), is defined as

$$\begin{aligned}
\underset{\boldsymbol{\theta}_{\mathcal{R}^1},\ldots,\boldsymbol{\theta}_{\mathcal{R}^{n_{so}}},\mathbf{z}}{\text{minimize}} \quad & \sum_{a\in\mathcal{M}} C_a(\boldsymbol{\theta}_{\mathcal{R}^a}) && (P_D)\\
\text{subject to} \quad & \boldsymbol{\theta}_{\mathcal{R}^a} \in \Theta_{\mathcal{R}^a},\ a \in \mathcal{M}, && (P_D\text{-}1)\\
& \mathbf{z} \in \Theta, && (P_D\text{-}2)\\
& \boldsymbol{\theta}_{\mathcal{R}^a} - \mathbf{z}_{\mathcal{R}^a} = \mathbf{0}, a \in \mathcal{M}. && (P_D\text{-}3)
\end{aligned}$$

The decision variables of Problem (P_D) are the voltage angles of all SOs, $\boldsymbol{\theta}_{\mathcal{R}^1}, \ldots, \boldsymbol{\theta}_{\mathcal{R}^{n_{so}}}$, and the global voltage angle variable, $\mathbf{z} \in \mathbb{R}^{n_b}$. Each SO$_a$'s decision variables are restricted to lie within the local constraint set $\Theta_{\mathcal{R}^a}$, by constraint (P_D-1), and $\mathbf{z} \in \Theta$, by constraint (P_D-2). We impose the coupling constraints (P_D-3), $\boldsymbol{\theta}_{\mathcal{R}^a} - \mathbf{z}_{\mathcal{R}^a} = \mathbf{0}$ for each SO$_a$, which states SO$_a$'s proposal must agree with the relevant components from the global variable, denoted by $\mathbf{z}_{\mathcal{R}^a}$.

3.4. Solution Methodology

The proposed solution method for Problem (P_D) consists of an iterative message exchange process which makes use of the ADMM algorithm (see [Boyd et al., 2011], original work [Glowinski & Marroco, 1975, Gabay & Mercier, 1976, Gabay, 1983]). Due to the structure of our problem, the conditions for convergence of the message exchange process are naturally met. Furthermore, under the assumption that c_n's are strictly increasing (assumption 3), we are able to ensure convergence of the optimizers of Problem (P_D) to those of the centralized information problem, Problem (P_C'), and consequently, obtain the set of optimal net injections, $\boldsymbol{I}_C^*$, for the centralized problem (P_C).

3.4.1. Message Exchange Process

We first form the partial augmented Lagrangian corresponding to Problem (P_D) by dualizing the coupling constraints (P_D-3) as

$$\mathcal{L}_\mu(\boldsymbol{\theta}_{\mathcal{R}^1},\dots,\boldsymbol{\theta}_{\mathcal{R}^{n_{so}}},\mathbf{z},\mathbf{y}) = \sum_{a\in\mathcal{M}} \Big(C_a(\boldsymbol{\theta}_{\mathcal{R}^a}) + \mathbf{y}_{\mathcal{R}^a}^\top(\boldsymbol{\theta}_{\mathcal{R}^a}-\mathbf{z}_{\mathcal{R}^a}) + \frac{\mu}{2}||\boldsymbol{\theta}_{\mathcal{R}^a}-\mathbf{z}_{\mathcal{R}^a}||_2^2\Big)$$

where μ is termed the *penalty parameter.* By the ADMM algorithm, primal variables are updated in parallel, by each SO, via

$$\begin{aligned}\boldsymbol{\theta}_{\mathcal{R}^a}^{t+1} &= \operatorname*{argmin}_{\boldsymbol{\theta}_{\mathcal{R}^a}\in\Theta_{\mathcal{R}^a}} \mathcal{L}_\mu(\boldsymbol{\theta}_{\mathcal{R}^1},\dots,\boldsymbol{\theta}_{\mathcal{R}^{n_{so}}},\mathbf{z}^t,\mathbf{y}^t)\\ &= \operatorname*{argmin}_{\boldsymbol{\theta}_{\mathcal{R}^a}\in\Theta_{\mathcal{R}^a}}\Big(C_a(\boldsymbol{\theta}_{\mathcal{R}^a}) + \mathbf{y}_{\mathcal{R}^a}^{t\top}\boldsymbol{\theta}_{\mathcal{R}^a} + \frac{\mu}{2}\Big\|\boldsymbol{\theta}_{\mathcal{R}^a}-\mathbf{z}_{\mathcal{R}^a}^t\Big\|_2^2\Big)\end{aligned}$$

followed by

$$\begin{aligned}\mathbf{z}^{t+1} &= \operatorname*{argmin}_{\mathbf{z}\in\Theta} \mathcal{L}_\mu(\boldsymbol{\theta}_{\mathcal{R}^1}^{t+1},\dots,\boldsymbol{\theta}_{\mathcal{R}^{n_{so}}}^{t+1},\mathbf{z},\mathbf{y}^t)\\ &= \operatorname*{argmin}_{\mathbf{z}\in\Theta} \sum_{a\in\mathcal{M}}\Big(-\mathbf{y}_{\mathcal{R}^a}^{t\top}\mathbf{z}_{\mathcal{R}^a} + \frac{\mu}{2}\Big\|\boldsymbol{\theta}_{\mathcal{R}^a}^{t+1}-\mathbf{z}_{\mathcal{R}^a}\Big\|_2^2\Big).\end{aligned}$$

Lastly, the dual variables for each $m \in \mathcal{M}$ are updated as

$$\mathbf{y}_{\mathcal{R}^a}^{t+1} = \mathbf{y}_{\mathcal{R}^a}^t + \mu\Big(\boldsymbol{\theta}_{\mathcal{R}^a}^{t+1} - \mathbf{z}_{\mathcal{R}^a}^{t+1}\Big). \tag{3.10}$$

It can be shown [Boyd et al., 2011] that the dual variables have a zero sum after the first iteration resulting in the $\mathbf{z}$ update reducing to an averaging of the elements of the elements of $\boldsymbol{\theta}_{\mathcal{R}^a}^{t+1}$. Each SO does this averaging locally and thus does not require a centralized entity (details found in Algorithm 1). Due to the convexity of the feasible sets $\Theta_{\mathcal{R}^a}$ for all $a \in \mathcal{M}$ and the fact that adjacent SOs, a and b, share common information of the line limits of their inter-SO$_{ab}$ network, the averaged vector $\mathbf{z}_{\mathcal{R}^a}$ also lies within $\Theta_{\mathcal{R}^a}$.

Algorithm 1 Message Exchange Process

```
Initialize $t = 0$, choose $\boldsymbol{\theta}^0_{\mathcal{R}^a}$, $\mathbf{y}^0_{\mathcal{R}^a}$ for $a \in \mathcal{M}$, $\mathbf{z}^0$, and $\mu > 0$
while $\neg(||p_r^{(t)}||_2 < \varepsilon_{\text{primal}}$ and $||d_r^{(t)}||_2 < \varepsilon_{\text{dual}})$ do
    for ( do (parallel optimization and broadcast)) $a \in \mathcal{M}$
        SO$_a$ solves:
            $$\boldsymbol{\theta}^{t+1}_{\mathcal{R}^a} = \underset{\boldsymbol{\theta}_{\mathcal{R}^a} \in \Theta_{\mathcal{R}^a}}{\operatorname{argmin}} \left( C_a(\boldsymbol{\theta}_{\mathcal{R}^a}) + \mathbf{y}^{t\top}_{\mathcal{R}^a} \boldsymbol{\theta}_{\mathcal{R}^a} + \frac{\mu}{2} \left\| \boldsymbol{\theta}_{\mathcal{R}^a} - \mathbf{z}^t_{\mathcal{R}^a} \right\|_2^2 \right)$$
        Broadcast $\theta_n^{(a),t+1}$ to SO$_{a'}$ for all $n \in \mathcal{R}^a \cap \mathcal{R}^{a'}$, $a' \in \mathcal{M}^a$.
    end for
    for ( do(parallel average and dual variable update)) $a \in \mathcal{M}$
        Average:
            $$z_n^{t+1} = \frac{1}{|\mathcal{M}^a| + 1} \left( \theta_n^{(a),t+1} + \sum_{a' \in \mathcal{M}^a} \theta_n^{(a'),t+1} \right)$$
        for all $n \in \mathcal{R}^a \cap \mathcal{R}^{a'}$ for all $a' \in \mathcal{M}^a$ and sets $z_n^{t+1} = \theta_n^{t+1}$ for all non-shared buses.
        Update: $\mathbf{y}^{t+1}_{\mathcal{R}^a} = \mathbf{y}^t_{\mathcal{R}^a} + \mu(\boldsymbol{\theta}^{t+1}_{\mathcal{R}^a} - \mathbf{z}^{t+1}_{\mathcal{R}^a})$
    end for
    Update residuals: compute $p_r^{(t+1)}$, $d_r^{(t+1)}$ via Eq.'s (3.11), (3.12)
    Update counter: $t \leftarrow t + 1$
end while
```

3.4.2. Algorithm Convergence

In order to establish convergence of the algorithm, we need to ensure that the unaugmented Lagrangian, $\mathcal{L}_0$, has a saddle point. First, we assume that Problem (P_D) satisfies Slater's condition, that is, the feasible set $\Theta_{\mathcal{R}^1} \times \cdots \times \Theta_{\mathcal{R}^{n_{so}}} \times \Theta$ has a nonempty interior (this assumption is reasonable in practical problems; the constraint set can be trivially modified to have a nonempty interior). As a result, by Cor. 28.3.1 of [Rockafellar, 1970] (p.283), the unaugmented Lagrangian, $\mathcal{L}_0$, has a saddle point. Furthermore, since each SO's aggregated cost function is convex by construction and closed (since each $\Theta_{\mathcal{R}^a}$ is compact), our problem is convex and satisfies the conditions required for the ADMM to converge ([Boyd et al., 2011], p.17).

The convergence result of [Boyd et al., 2011] ensures that the ADMM results in convergence of the primal residuals to zero (solution approaches feasibility), the objective function to the optimal value, and the dual variables to the optimal dual point. It does not, in general, ensure convergence of the primal variables to their optimal values. However, by Lemma 3.3.1, each SO's aggregated cost function C_a is strongly convex, ensuring that the sequence $\left\{(\boldsymbol{\theta}^t_{\mathcal{R}^1}, \ldots, \boldsymbol{\theta}^t_{\mathcal{R}^{n_{so}}}, \mathbf{z}^t)\right\}$ generated by Algorithm 1 converges to an optimal solution, as summarized by the following corollary.

Corollary 3.4.1. *The sequence $\{\mathbf{z}^t\}$ generated by Algorithm 1 converges to the unique optimal solution of Problem (P'_C).*

Using the sequence of angles $\{\mathbf{z}^t\}$ generated by Algorithm 1, we define the corresponding sequence of net power injections, denoted $\{\boldsymbol{I}^t_D\}$, where each term is defined as $\boldsymbol{I}^t_D = \left(f_1(\mathbf{z}^t_{\mathcal{R}_1}), \ldots, f_{n_b}(\mathbf{z}^t_{\mathcal{R}_{n_b}})\right)$, and state the following corollary.

Corollary 3.4.2. *The sequence of net injections $\{\boldsymbol{I}^t_D\}$ converges to $\boldsymbol{I}^*_C$, the unique optimal solution of Problem (P_C), and achieves the same optimal social cost, c^*_C.*

Generators and loads at each bus are then required to meet their respective buses' prescribed optimal net injection. The resulting set of injections induce tie-line flows which are consistent with the social-cost-minimizing solution under the modified DC approximation.

3.5. Numerical Examples

For purposes of simulation, we compute the primal and dual residual, which serve as optimality measures, and compare their norms to a fixed threshold to determine when to terminate the algorithm. The primal and dual residuals are defined respectively as

$$d_r^{(t)} := (\boldsymbol{\theta}^t_{\mathcal{R}^1} - \mathbf{z}^t_{\mathcal{R}^1}, \ldots, \boldsymbol{\theta}^t_{\mathcal{R}^{n_{so}}} - \mathbf{z}^t_{\mathcal{R}^{n_{so}}}), \tag{3.11}$$

$$p_r^{(t)} := -\mu(\mathbf{z}^t_{\mathcal{R}^1} - \mathbf{z}^{t-1}_{\mathcal{R}^1}, \ldots, \mathbf{z}^t_{\mathcal{R}^{n_{so}}} - \mathbf{z}^{t-1}_{\mathcal{R}^{n_{so}}}). \tag{3.12}$$

For specified thresholds, $\varepsilon_{\text{primal}}, \varepsilon_{\text{dual}} > 0$, we terminate the algorithm when $||d_r^{(t)}||_2 < \varepsilon_{\text{primal}}$ and $||p_r^{(t)}||_2 < \varepsilon_{\text{dual}}$.

We now demonstrate the performance of Algorithm 1 on two systems: (1) A 12 bus, 3-region system illustrated in Fig. 3.1; (2) A 73 bus, 3-region system (IEEE RTS-96 system). For the 12 bus system, buses are classified as $\mathcal{N}_b^g = \{4, 8, 9, 12\}$, $\mathcal{N}_b^d = \{1, 6, 7, 10\}$, with remaining buses of hybrid type. Slack bus indices are $\mathcal{N}_b^s = \{3, 5\}$. Injection bounds are $p_n^{\min} = 0\text{MW}$ for all $n \in \mathcal{N}_b^g$, $p_n^{\max} = 300\text{MW}$ for all $n \in \mathcal{N}_b^g$ and hybrid buses; $p_2^{\max} = -200\text{MW}$, $p_6^{\max} = -300\text{MW}$, $p_7^{\max} = -100\text{MW}$ (these buses must receive at least $p_n^{\max}$); $p_2^{\min} = -400\text{MW}$ (bus 2 cannot receive more than 400MW); and $p_n^{\min} = -\infty$ for all remaining consumption and hybrid buses. Stability bounds $\underline{\theta}_{nm}$ and $\overline{\theta}_{nm}$ are set at $\pm\pi/2$ and line limits are denoted by the parenthesized values on the lines in Fig. 3.1. Algorithm parameters are set as follows: initial conditions $\boldsymbol{\theta}^0_{\mathcal{R}^a} = \mathbf{0}$, $\mathbf{y}^0_{\mathcal{R}^a} = \mathbf{0}$ for $a \in \mathcal{M}$, $\mathbf{z}^0 = \mathbf{0}$; penalty parameter $\mu = 9\times10^{-5}$; and stopping thresholds $\varepsilon_{\text{primal}} = 2\times10^{-2}$ and $\varepsilon_{\text{dual}} = 1\times10^{-3}$. Fig. 3.1 and Table 3.1 present the convergent flows and injections, respectively, for the 12 bus system. Algorithm parameters the 73 bus system are: $\boldsymbol{\theta}^0_{\mathcal{R}^a} = \mathbf{0}$, $\mathbf{y}^0_{\mathcal{R}^a} = \mathbf{0}$ for $a \in \mathcal{M}$, $\mathbf{z}^0 = \mathbf{0}$, $\mu = 25$, $\varepsilon_{\text{primal}} = 1\times10^{-2}$, and $\varepsilon_{\text{dual}} = 2\times10^{-2}$. Fig. 3.2 shows convergence results for both the 12 bus and 73 bus systems. Cost functions for both systems take the form $c_n(p_n) = a_n \exp(p_n + b_n) + c_n$, where $a_n > 0$.

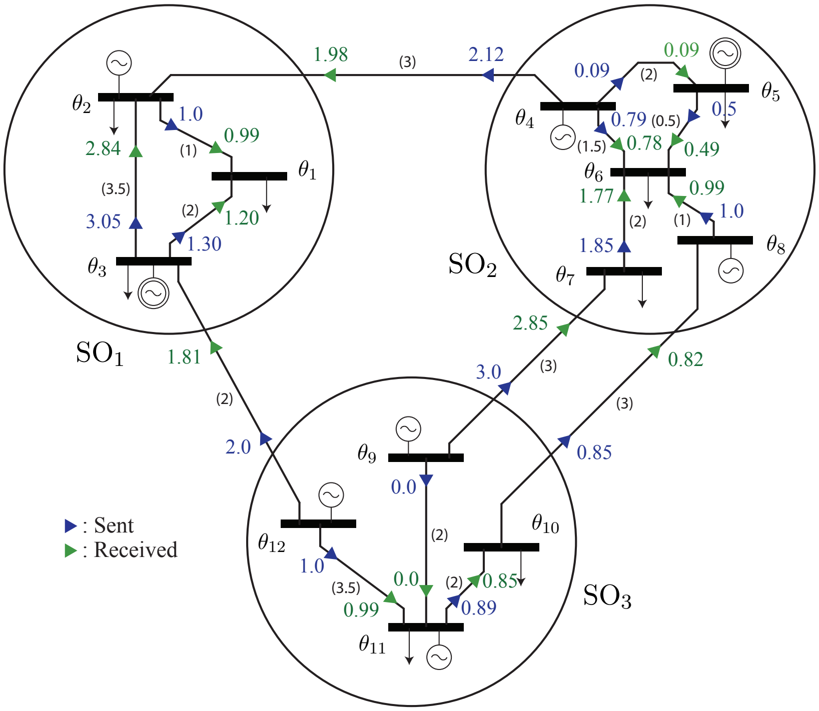


Figure 3.1: 12 bus example; SO's decision variables are $\boldsymbol{\theta}_{\mathcal{R}^1} = \{\theta_1, \theta_2^{(1)}, \theta_4^{(1)}, \theta_{12}^{(1)}\}$, $\boldsymbol{\theta}_{\mathcal{R}^2} = \{\theta_2^{(2)}, \theta_4^{(2)}, \theta_6, \theta_7^{(2)}, \theta_8^{(2)}, \theta_9^{(2)}, \theta_{10}^{(2)}\}$, and $\boldsymbol{\theta}_{\mathcal{R}^3} = \{\theta_7^{(3)}, \theta_8^{(3)}, \theta_9^{(3)}, \theta_{10}^{(3)}, \theta_{11}, \theta_{12}^{(3)}\}$; buses with double-encircled generators symbolize slack buses.

3.6. Discussion & Conclusion

As seen in Table 3.1, the convergent injections satisfy the injection constraints defined in Section 3.5. For example, the convergent net injection at bus 4 binds the upper bound constraint of $p_4^{\max} = 300$MW. Notice that this constraint is on the lossy net injection; this can be seen by observing the power flowing out of bus 4 in Fig. 3.1. Also notice that, due to losses, the power leaving from bus i to bus j is higher than the power received at bus j from bus i, and consequently, the line limit is on the loss-included flow (for example, in Fig. 3.1, the line limit on line $\{7, 9\}$ limits the power flowing from bus 9 to bus 7 to 300MW).

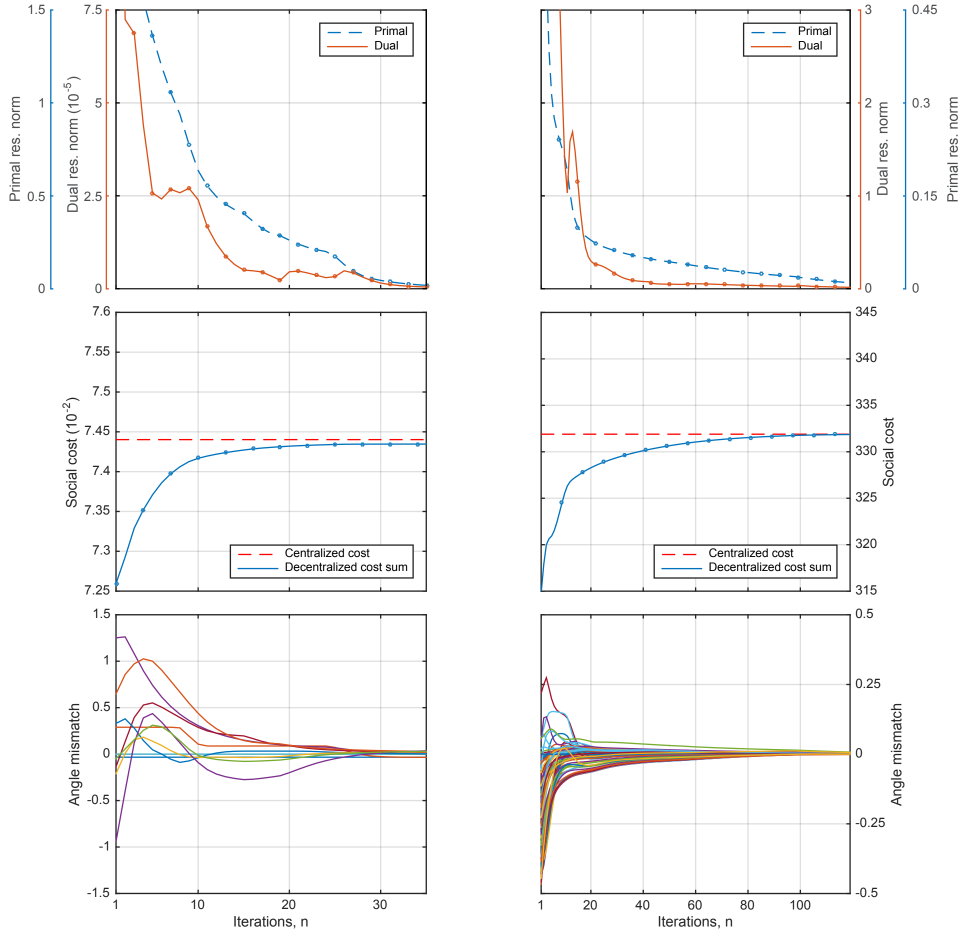
Bus	Injection	Bus	Injection	Bus	Injection
1	-220.17	5	41.36	9	300.00
2	-382.24	6	-403.70	10	0.00
3	253.86	7	-100	11	-11.00
4	300.00	8	17.80	12	300.00

Table 3.1: Convergent injections (in MW) for the 12 bus system.

The algorithm has been shown to converge quickly to the centralized optimum. Convergence to the specified tolerances was achieved in a relatively small number of iterations; 35 iterations for the 12 bus system and 119 iterations in the 73 bus system.¶ As discussed in [Boyd et al., 2011] the ADMM behaves much like a first-order method, in the sense that it can be slow to converge to high-accuracy; however, moderate accuracy can be obtained in the order of tens of iterations. This behavior was confirmed by the simulations performed on the test systems. The strong convexity of each SO's cost function was found to prevent oscillation between optimal solutions, resulting in faster convergence (observed through simulation results). Empirically, we have observed that the speed of convergence is heavily influenced by the choice of penalty parameter μ; poorly chosen values of μ can result in slow convergence.

In summary, the proposed algorithm obtains a cost-minimizing solution which satisfies the physical constraints of the system while obeying the informational constraints of the SOs. The process involves SOs exchanging voltage angle messages with their neighbors, eventually reaching an agreed-upon set of angles. The convergent angles define power flows between SOs and a corresponding set of cost-minimizing net injections for the system.

¶Small loads were placed at the zero injection buses in the 73 bus test system.



(a) 12-bus example

(b) 73-bus example

Figure 3.2: Convergence plots for **(a)** the 12 bus (first column) and **(b)** the 73 bus (second column) systems; primal and dual residual norms (top), see Eq.'s (4.6) and (4.7); centralized and decentralized cost sum (middle); and angle mismatch, $\theta_n^* - z_n^t$, $i \in \mathcal{N}$, between the centralized optimal $\boldsymbol{\theta}^*$ and $\mathbf{z}^t$ (bottom), as functions, n.

Chapter 4

A Decentralized Mechanism for Computing Competitive Equilibria in Deregulated Electricity Markets

There are some drawbacks of the approach taken in Chapter 3 that motivates the development of a more realistic model. In Chapter 3, the requirement to have a convex problem (in order to ensure convergence of the ADMM algorithm), results in some unrealistic assumptions, such as placing a lower bound on the non-lossy injection and not being able to consider zero-injection buses. The approach of Chapter 4 does not require convexity in order to ensure convergence, permitting a more realistic model.

4.1. Introduction

From the introduction of the Public Utilities Regulatory Policies Act (PURPA) in 1978 to the establishment of the Energy Policy Act in 1992, the deregulation of electricity markets in the United States has grown continuously, primarily under the appeal of increased technological competition and innovation. Today, despite cases of market manipulation (such as the California electricity crisis in 2000-2001), many large electricity markets are, at least in some capacity, deregulated. This transition has been centered around the formation of specialized firms for generation, transmission, and distribution, to name a few, with markets typically consisting of the following companies [Christie & Bose, 1996] (termed *market participants*): *generation companies* (GenCos) who produce and sell power, *transmission companies* (TransCos) who own the transmission assets and are responsible for transmitting power across the grid, and *distribution companies* (DistCos) who own the distribution networks and are tasked with buying power from GenCos and distributing it to consumers. The primary goal in an electricity market is determining an outcome that is not only economically *optimal* (that is, it is *Pareto efficient* [Mas-Colell et al., 1995, Kirschen & Strbac, 2004]) but also satisfies the physical constraints of the system.

Centralized market mechanisms are traditionally the approach used for determining the optimal, feasible outcome of the market [Stoft, 2002]. Under these approaches a centralized market operator receives *bids* from the market participants, in the form of cost/benefit functions and technical constraints, and solves a large-scale centralized optimization problem to determine the market clearing outcome. This outcome consists of a physically feasible operating point as well as a vector of bus-specific power prices termed *locational marginal prices* (LMPs). Unfortunately, centralized mechanisms suffer from some drawbacks. First, reporting cost and technical information raises privacy concerns for market participants. Also, as systems grow in size, the centralized optimization problem can become prohibitively large.

This issue is made worse by the recent surge in distributed generation and demand side participation [Papadaskalopoulos & Strbac, 2013], further increasing the dimensionality and complexity of the problem and potentially making centralized mechanisms computationally intractable.

In hopes of avoiding these drawbacks, we introduce a decentralized market mechanism which achieves the economically optimal outcome, honoring the informational asymmetries of the problem and considering important nonlinearities of the system (such as power losses and limits on transmission lines). The electricity market model consists of multiple market participants, DistCos, GenCos, and TransCos, and a single market operator. Our model allows for the consumption centers of each DistCo and the production centers of each GenCo to be distributed across the network. For example, a given GenCo could own generators at multiple buses in the network (a portfolio of plants). Additionally, our model allows for the ownership of transmission lines in the system to be partitioned among multiple TransCos. The market operator is responsible for obtaining a market clearing outcome. The process of achieving this market clearing outcome, termed a *decentralized market mechanism*, is based on principles from Lagrangian duality theory, specifically making use of the *dual decomposition* method [Bertsekas, 1999]. The mechanism, which we refer to as the *pricing process*, consists of an iterative price response and price update procedure. All market participants are assumed to act in a self-optimizing manner, that is, given the current LMPs they adjust their decision variables in order to maximize their financial surplus subject to their own local physical and operational constraints. This allows, for instance, for DistCos to exercise flexible demand participation for the elastic component of their total demand and for GenCos to self-dispatch. DistCos and GenCos optimize independently, reporting their surplus-maximizing consumption and production profiles, respectively. TransCos partake in a cooperative message exchange process to reach an operating point that induces power flows that maximize their surpluses for transmitting power along their respective lines. The optimizers are sent to the market operator who is responsible for updating the LMPs in such a way that the self-interested behavior of market participants leads to an outcome that is physically feasible. This outcome, when coupled with the associated set of LMPs, forms a *competitive equilibrium* [Mas-Colell et al., 1995, Motto et al., 2002a], which we show is Pareto efficient. Under relatively weak conditions (a convex DC approximation and edge-wise positive sums of LMPs), the market participants' optimization problems are convex and the pricing process converges. The pricing process avoids the need for market participants to reveal sensitive information, and additionally, the mechanism scales much more effectively than its centralized counterpart.

4.1.1. Literature Review

We focus on papers from the literature that are most similar to ours, primarily including works that develop decentralized market mechanisms (under perfect competition) using Lagrangian duality techniques. Duality theory allows one to solve the computationally simpler dual problem; however, this can result in a non-zero *duality gap* in general. The authors of [Motto et al., 2002a, Galiana et al., 2002] construct a market model consisting of GenCos, DistCos, and a single TransCo, considering a fully nonlinear AC power flow model. Their decentralized mechanism, based on a dual approach, is conjectured to converge to a zero duality gap solution under *profit optimality* and a *convexifying market rule* (a restriction of market participants' behavior). Lavaei and Sojoudi [Lavaei & Sojoudi, 2012] consider a competitive energy market setting with GenCos, DistCos, and an ISO under the AC power flow model (using an SDP reformulation). Assuming positive LMPs, the authors are able to show convergence to a zero duality gap solution under the assumption of either: a radial network, or, in the case of a mesh network, the existence of a *phase-shifter* for each network cycle. In the absence of phase-shifters, a zero duality gap can be ensured if loads are allowed to be *over-satisfied* (discarding extra power). Similar mechanisms have been applied in the context of the unit commitment problem (e.g. [Zhuang & Galiana, 1988], [Bard, 1988], [Ongsakul & Petcharaks, 2004]) and demand response exchange markets (see [Nguyen et al., 2012] and [Papadaskalopoulos & Strbac, 2013]).

4.1.2. Contribution

The contributions of this chapter are twofold:

1) ***Modeling generality***: Our model allows for the ownership of power system assets to be partitioned among the market participants. This allows for each DistCo and GenCo to own multiple units that are distributed across the network (existing literature assumes that each participant owns a single unit [Motto et al., 2002a, Galiana et al., 2002, Lavaei & Sojoudi, 2012]). Our model also allows for ownership of lines to be partitioned among multiple TransCos ([Motto et al., 2002a, Galiana et al., 2002] consider a single TransCo).

2) ***Convergence to a zero duality gap solution***: Existing models contain nonlinearities that either preclude convergence guarantees [Motto et al., 2002a, Galiana et al., 2002] or require strong sufficient conditions [Lavaei & Sojoudi, 2012]. Our model allows us to ensure convergence (under natural conditions) while preserving important nonlinearities of the problem, such as power losses.

4.2. Energy Market Model

In addition to the market operator (MO), the market model of this chapter contains three types of agents (market participants): DistCos, denoted by the set $\mathcal{DC} = \{1, \ldots, D_c\}$; GenCos, denoted by $\mathcal{GC} = \{1, \ldots, G_c\}$; and TransCos, denoted by $\mathcal{TC} = \{1, \ldots, T_c\}$. Each DistCo $i \in \mathcal{DC}$ owns *consumption units*, consisting of elastic loads at buses $\mathcal{N}^i_{\mathcal{DC}^d} \subseteq \mathcal{N}$ and inelastic loads at buses $\mathcal{N}^i_{\mathcal{DC}^s} \subseteq \mathcal{N}$. The elastic and inelastic load profiles of DistCo $i \in \mathcal{DC}$ are $\mathbf{d}^i = \left\{d^i_n\right\}_{n \in \mathcal{N}^i_{\mathcal{DC}^d}}$ and $\mathbf{s}^i = \left\{s^i_n\right\}_{n \in \mathcal{N}^i_{\mathcal{DC}^s}}$, respectively, where $d^i_n \in [\underline{d}^i_n, \bar{d}^i_n]$ is the elastic demand and $s^i_n \geq 0$ is the (given) inelastic demand of DistCo i's consumption unit at bus n. Each GenCo $i \in \mathcal{GC}$ owns *generation units* at buses $\mathcal{N}^i_{\mathcal{GC}} \subseteq \mathcal{N}$. The real power injection profile of GenCo $i \in \mathcal{GC}$ is $\mathbf{p}^i = \left\{p^i_n\right\}_{n \in \mathcal{N}^i_{\mathcal{GC}}}$ where $p^i_n \in [\underline{p}^i_n, \bar{p}^i_n]$ is the injection of GenCo i's generation unit at bus n. For convenience, let $p^i_n = 0$ if GenCo i does not own a generation unit at bus n (similarly for d^i_n, s^i_n of DistCo i at bus n). Lastly, each TransCo $i \in \mathcal{TC}$ owns a set of transmission lines $\mathcal{E}^i_l$ with ownership of lines in the system partitioned among TransCos, that is, $\mathcal{E}^1_l \cup \cdots \cup \mathcal{E}^{T_c}_l = \mathcal{E}_l$ and $\mathcal{E}^i_l \cap \mathcal{E}^j_l = \varnothing$, $i \neq j$. Each edge-set $\mathcal{E}^i_l$ has an associated set of buses $\mathcal{N}^i_{\mathcal{TC}}$ defined as the endpoints of the edges in $\mathcal{E}^i_l$. The associated voltage angle profile of TransCo $i \in \mathcal{TC}$ is $\boldsymbol{\theta}^i = \left\{\theta_n\right\}_{n \in \mathcal{N}^i_{\mathcal{TC}}}$. For later convenience, we also define $\mathcal{N}^{i,j}_{\mathcal{TC}} := \mathcal{N}^i_{\mathcal{TC}} \cap \mathcal{N}^j_{\mathcal{TC}}$ as the set of shared buses between two TransCos' edge-sets $\mathcal{E}^i_l$ and $\mathcal{E}^j_l$ and $\mathcal{TC}_n := \{i \in \mathcal{TC} \mid n \in \mathcal{N}^i_{\mathcal{TC}}\}$ as the set of TransCos that own lines that are connected to bus n. A sample network can be seen in Fig. 4.1.

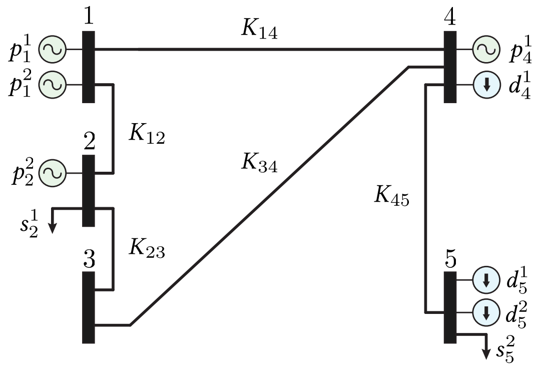


Figure 4.1: A sample 5-bus network. GenCo $i = 1$ owns generator units at buses $\mathcal{N}^1_{\mathcal{GC}} = \{1, 4\}$ corresponding to an injection vector $\mathbf{p}^1 = (p^1_1, p^1_4)$. GenCo $i = 2$ has generator units at buses $\mathcal{N}^2_{\mathcal{GC}} = \{1, 2\}$, $\mathbf{p}^2 = (p^2_1, p^2_2)$; DistCo $i = 1$ has elastic loads at buses 4 and 5, $\mathbf{d}^1 = (d^1_4, d^1_5)$, and an inelastic load at bus 2, $\mathbf{s}^1 = s^1_2$, thus $\mathcal{N}^1_{\mathcal{DC}^d} = \{4, 5\}$, $\mathcal{N}^1_{\mathcal{DC}^s} = \{2\}$; and lastly, DistCo $i = 2$ has both an elastic and inelastic load at bus $\mathcal{N}^2_{\mathcal{DC}^d} = \mathcal{N}^2_{\mathcal{DC}^s} = \{5\}$, thus $\mathbf{d}^2 = d^2_5$, $\mathbf{s}^2 = s^2_5$. Bus 3 is a zero-injection bus. TransCo $i = 1$ owns lines $\mathcal{E}^1_l = \{\{1, 2\}, \{1, 4\}, \{2, 3\}\}$ thus $\mathcal{N}^1_{\mathcal{TC}} = \{1, 2, 3, 4\}$ and TransCo $i = 2$ owns lines $\mathcal{E}^2_l = \{\{3, 4\}, \{4, 5\}\}$ so $\mathcal{N}^2_{\mathcal{TC}} = \{3, 4, 5\}$.

The load and generation profiles of DistCos and GenCos have associated utilities and costs, respectively. For an elastic load profile $\mathbf{d}^i$ the aggregate utility (benefit) function of

DistCo i is defined as $\mathbf{u}^i\left(\mathbf{d}^i\right) := \sum_{n\in\mathcal{N}^i_{\mathcal{DC}^d}} u^i_n\left(d^i_n\right)$, where $u^i_n(d^i_n)$ is the benefit associated with elastic demand level d^i_n. Similarly, GenCo i's aggregate cost function (total generation cost) is $\mathbf{c}^i\left(\mathbf{p}^i\right) := \sum_{n\in\mathcal{N}^i_{\mathcal{GC}}} c^i_n\left(p^i_n\right)$ where $c^i_n(p^i_n)$ represents the cost for producing real power p^i_n.

4.2.1. Model Assumptions

In addition to the convex power flow approximation, introduced in Section 2.2, we impose the following four modeling assumptions for this chapter.

Assumption 1 *(slack buses)*: Denote the set of slack buses by $\mathcal{N}^s_b$. We require that each TransCo has exactly one slack bus, that is, $\mathcal{N}^i_{\mathcal{TC}} \cap \mathcal{N}^s_b$ contains one element for all $i \in \mathcal{TC}$. Slack buses serve solely as angle references, that is, $\theta_n = 0$ for all $n \in \mathcal{N}^s_b$.

Assumption 2 *(strong convexity)*: We require that all DistCo utility functions u^i_n are strongly concave and all GenCo cost functions c^i_n are strongly convex (this condition is equivalent to strict convexity if the functions are quadratic).

Assumption 3 *(positive edge-wise sums of prices)*: We require that all edge-wise sums of locational marginal prices are positive. That is, $\lambda_n + \lambda_m > 0$ for all $\{n, m\} \in \mathcal{E}_l$.[*] Note that this allows $\lambda_n < 0$ for some n.

Assumption 4 *(price-taking behavior)*: We assume that the agents (market participants) are price-taking, that is, they assume that the price will remain unchanged if they change their response. This requires that agents are non-strategic, that is, they obey the rules of the mechanism and do not need to be incentivized to participate.

4.2.2. Knowledge Model

We now describe the assumptions regarding information in our problem. Each DistCo $i \in \mathcal{DC}$ possesses private information regarding their utility functions $\{u^i_n\}_{n\in\mathcal{N}^i_{\mathcal{DC}^d}}$ and any

[*] Note: We are not enforcing this as a constraint in our problem, rather we are only considering topologies where this assumption is naturally satisfied.

bounds on the elastic load level $\underline{\mathbf{d}}^i = \{\underline{d}_n^i\}_{n\in\mathcal{N}^i_{\mathcal{DC}^d}}, \bar{\mathbf{d}}^i = \{\bar{d}_n^i\}_{n\in\mathcal{N}^i_{\mathcal{DC}^d}}$. Each GenCo $i \in \mathcal{GC}$ possesses private information regarding their cost functions $\{c_n^i\}_{n\in\mathcal{N}^i_{\mathcal{GC}}}$ and production bounds $\underline{\mathbf{p}}^i = \{\underline{p}_n^i\}_{n\in\mathcal{N}^i_{\mathcal{GC}}}, \bar{\mathbf{p}}^i = \{\bar{p}_n^i\}_{n\in\mathcal{N}^i_{\mathcal{GC}}}$. Each TransCo $i \in \mathcal{TC}$ knows the connectivity of their region of the network, $(\mathcal{N}^i_{\mathcal{TC}}, \mathcal{E}^i_l)$, as well as the admittances of the corresponding lines, Y_{nm} for $\{n, m\} \in \mathcal{E}^i_l$. TransCos also possess private information of the line limits of their transmission lines, K_{nm}, $\{n, m\} \in \mathcal{E}^i_l$. Each DistCo $i \in \mathcal{DC}$ knows the inelastic demands at its buses, $\{s_n^i\}_{n\in\mathcal{N}^i_{\mathcal{DC}}}$, whereas the MO is assumed to know all inelastic demand levels. Furthermore, the MO knows the location of all DistCo and GenCo units, the network connectivity, and the admittances of all transmission lines in the network.

4.3. Maximizing Social Welfare

We are interested in determining the set of variables, consisting of DistCo elastic demand levels $\{\mathbf{d}^i\}_{i\in\mathcal{DC}}$, GenCo real power injection levels $\{\mathbf{p}^i\}_{i\in\mathcal{GC}}$, and an operating point $\boldsymbol{\theta}$, such that the *social welfare* is maximized, subject to physical and operational constraints. It is known from microeconomic theory that maximizing the social welfare results in a Pareto efficient outcome [Mas-Colell et al., 1995]. The single time-period problem can be formally stated as Problem (P) below.

$$\max_{\mathbf{x}=(\{\mathbf{d}^i\}_{i\in\mathcal{DC}},\{\mathbf{p}^i\}_{i\in\mathcal{GC}},\boldsymbol{\theta})} W(\mathbf{x}) := \sum_{i\in\mathcal{DC}} \mathbf{u}^i\left(\mathbf{d}^i\right) - \sum_{i\in\mathcal{GC}} \mathbf{c}^i\left(\mathbf{p}^i\right) \tag{P}$$

$$\text{s.t. } \mathbf{p} - (\mathbf{d} + \mathbf{s}) = \boldsymbol{f}\left(\boldsymbol{\theta}\right) \tag{P.i}$$

$$\underline{\mathbf{p}}^i \leq \mathbf{p}^i \leq \bar{\mathbf{p}}^i, i \in \mathcal{GC} \tag{P.ii}$$

$$\underline{\mathbf{d}}^i \leq \mathbf{d}^i \leq \bar{\mathbf{d}}^i, i \in \mathcal{DC} \tag{P.iii}$$

$$g\left(\theta_{nm}\right) \leq K_{nm}, g\left(\theta_{mn}\right) \leq K_{mn}, \{n, m\} \in \mathcal{E}_l \tag{P.iv}$$

$$\underline{\theta}_{nm} \leq \theta_{nm} \leq \overline{\theta}_{nm}, \{n, m\} \in \mathcal{E}_l \tag{P.v}$$

$$\theta_n = 0, n \in \mathcal{N}_b^s \tag{P.vi}$$

$$\theta_n \in [-\pi, \pi], n \in \mathcal{N}_b \tag{P.vii}$$

The objective function of Problem (P), $W(\mathbf{x})$, represents the *social welfare* and can be written as the total utility to DistCos minus the total cost to GenCos.[†]

The constraints of problem (P) arise from both physical laws and the operational requirements of the power system and the agents. The first constraint (P.i), termed the *power balance equation*, takes the form

$$\mathbf{p} - (\mathbf{d} + \mathbf{s}) = \boldsymbol{f}(\boldsymbol{\theta}) \tag{4.1}$$

where $\mathbf{p} = (p_1, \ldots, p_{n_b})$, with $p_n = \sum_{i\in\mathcal{GC}} p_n^i$, is the net generation vector and $(\mathbf{d}+\mathbf{s})$ is the net demand vector consisting of two components, the elastic demand vector $\mathbf{d} = (d_1, \ldots, d_{n_b})$

[†] The reason for this form is as follows. The social welfare is defined as the sum of agents' surplus functions, that is (using the notation of Section 4.4.1), $W(\mathbf{x}) = \sum_{i\in\mathcal{DC}} \Psi_{\mathcal{DC}}^i(\mathbf{d}^i, \boldsymbol{\lambda}) + \sum_{i\in\mathcal{GC}} \Psi_{\mathcal{GC}}^i(\mathbf{p}^i, \boldsymbol{\lambda}) + \sum_{i\in\mathcal{TC}} \Psi_{\mathcal{TC}}^i(\boldsymbol{\theta}^i, \boldsymbol{\lambda}) = \sum_{i\in\mathcal{DC}} \Psi_{\mathcal{DC}}^i(\mathbf{d}^i, \boldsymbol{\lambda}) + \sum_{i\in\mathcal{GC}} \Psi_{\mathcal{GC}}^i(\mathbf{p}^i, \boldsymbol{\lambda}) + \Psi_{\mathcal{TC}}(\boldsymbol{\theta}, \boldsymbol{\lambda})$. After substitution and rearrangement, $W(\mathbf{x}) = \sum_{i\in\mathcal{DC}} \left(\sum_{n\in\mathcal{N}_{\mathcal{DC}^d}^i}\left[u_n^i\left(d_n^i\right) - \lambda_n d_n^i\right] - \sum_{n\in\mathcal{N}_{\mathcal{DC}^s}^i} \lambda_n s_n^i\right) + \sum_{i\in\mathcal{GC}} \sum_{n\in\mathcal{N}_{\mathcal{GC}}^i} \left[\lambda_n p_n^i - c_n^i\left(p_n^i\right)\right] - \sum_{n\in\mathcal{N}_b} \lambda_n f_n(\boldsymbol{\theta}) = \sum_{i\in\mathcal{DC}} \mathbf{u}^i\left(\mathbf{d}^i\right) - \sum_{i\in\mathcal{GC}} \mathbf{c}^i\left(\mathbf{p}^i\right) + \lambda_n(p_n - (d_n + s_n) - f_n(\boldsymbol{\theta}))$. Applying the power balance equation, constraint (P.i), results in the desired expression.

and the (fixed) inelastic demand vector $\mathbf{s} = (s_1, \ldots, s_{n_b})$ (with $d_n = \sum_{i \in \mathcal{DC}} d_n^i$ and $s_n = \sum_{i \in \mathcal{DC}} s_n^i$). The vector $\boldsymbol{f}(\boldsymbol{\theta}) = \big(f_1(\boldsymbol{\theta}), \ldots, f_{n_b}(\boldsymbol{\theta})\big)$ denotes the power injections induced by the operating point $\boldsymbol{\theta}$, where the injection at bus n is defined by the convex function $f_n(\boldsymbol{\theta}) = \sum_{m \in \mathcal{N}_b} g(\theta_{nm})$, where $g(\theta_{nm})$ represents the power flow from bus n to m defined in Eq. (2.3); notice that $g(\theta_{nm})$ is zero if $\{n, m\} \notin \mathcal{E}_l$. Constraint (P.i) simply states that the injections due to physical laws, $\boldsymbol{f}(\boldsymbol{\theta})$, must agree with the net generation and demand at every bus. Constraints (P.ii) and (P.iii) reflect the fact that GenCos/DistCos have bounds on the amount of power they are able to produce/consume. Transmission constraints on the amount of power flowing on each line, constraint (P.iv), stability constraints on the voltage angle difference, (P.v), and slack references, (P.vi), are also imposed. The last constraint, (P.vii), is a technical condition that ensures that the voltage angles are well-defined. We group constraints (P.ii)-(vii) into a set denoted by $\mathbf{X}$. It is clear that $\mathbf{X}$ is convex since it is the intersection of half-spaces and convex inequality constraints. Lastly, we assume that Problem (P) is feasible.

There are some fundamental difficulties in obtaining a solution to Problem (P). First, the problem is nonconvex due the presence of the nonlinear power balance equation. Furthermore, by the discussion in Section 4.2.2, no single entity in the system has the information required to obtain a solution to Problem (P). The remainder of the chapter will focus on obtaining a solution to Problem (P).

4.4. Surpluses & Competitive Equilibria

The notion of a *competitive equilibrium* will be of central importance in obtaining a solution to Problem (P). Before we formally define a competitive equilibrium in the context of our problem, we need to discuss some aspects related to the Lagrangian dual function of Problem (P).

A partial Lagrangian of Problem (P) is formed by dualizing the power balance equation through the vector of dual variables, $\boldsymbol{\lambda}$, where each component λ_n represents the locational marginal price of power at bus n. Denoting the vector of variables by $\mathbf{x} = (\{\mathbf{d}^i\}_{i\in\mathcal{DC}}, \{\mathbf{p}^i\}_{i\in\mathcal{GC}}, \boldsymbol{\theta})$, and defining $\mathbf{h}(\mathbf{x}) := \boldsymbol{f}(\boldsymbol{\theta}) - \mathbf{p} + \mathbf{d} + \mathbf{s}$, the Lagrangian is

$$\begin{aligned}
\mathcal{L}\Big(\mathbf{x}, \boldsymbol{\lambda}\Big) :&= W(\mathbf{x}) - \boldsymbol{\lambda}^\top \mathbf{h}(\mathbf{x}) \\
&= \sum_{i\in\mathcal{DC}} \mathbf{u}^i\Big(\mathbf{d}^i\Big) - \sum_{i\in\mathcal{GC}} \mathbf{c}^i\Big(\mathbf{p}^i\Big) - \boldsymbol{\lambda}^\top(\boldsymbol{f}(\boldsymbol{\theta}) - \mathbf{p} + \mathbf{d} + \mathbf{s}) \\
&= \sum_{i\in\mathcal{DC}} \sum_{n\in\mathcal{N}^i_{\mathcal{DC}^d}} u^i_n\Big(d^i_n\Big) - \sum_{i\in\mathcal{GC}} \sum_{n\in\mathcal{N}^i_{\mathcal{GC}}} c^i_n\Big(p^i_n\Big) \\
&\quad - \sum_{n\in\mathcal{N}_b} \lambda_n \left(f_n(\boldsymbol{\theta}) - \sum_{i\in\mathcal{GC}} p^i_n + \sum_{i\in\mathcal{DC}} \Big(d^i_n + s^i_n\Big) \right) \\
&= \sum_{i\in\mathcal{DC}} \left(\sum_{n\in\mathcal{N}^i_{\mathcal{DC}^d}} \Big[u^i_n\Big(d^i_n\Big) - \lambda_n d^i_n\Big] - \sum_{n\in\mathcal{N}^i_{\mathcal{DC}^s}} \lambda_n s^i_n \right) \\
&\quad + \sum_{i\in\mathcal{GC}} \left(\sum_{n\in\mathcal{N}^i_{\mathcal{GC}}} \Big[\lambda_n p^i_n - c^i_n\Big(p^i_n\Big)\Big] \right) - \sum_{n\in\mathcal{N}_b} \lambda_n f_n(\boldsymbol{\theta}). \qquad (4.2)
\end{aligned}$$

Due to the structure of the Lagrangian, Eq. (4.2), the evaluation the dual function, defined as $\phi(\boldsymbol{\lambda}) = \max_{\mathbf{x}\in\mathbf{X}} \Big\{\mathcal{L}\Big(\mathbf{x}, \boldsymbol{\lambda}\Big)\Big\}$, is greatly simplified via separable optimizations.

$$\begin{aligned}
\phi(\boldsymbol{\lambda}) &= \max_{\mathbf{x}\in\mathbf{X}} \Big\{\mathcal{L}\Big(\mathbf{x}, \boldsymbol{\lambda}\Big)\Big\} \\
&= \sum_{i\in\mathcal{DC}} \max_{\mathbf{d}^i\in\mathbf{D}^i} \left\{ \sum_{n\in\mathcal{N}^i_{\mathcal{DC}^d}} \Big[u^i_n\Big(d^i_n\Big) - \lambda_n d^i_n\Big] - \sum_{n\in\mathcal{N}^i_{\mathcal{DC}^s}} \lambda_n s^i_n \right\} \\
&\quad + \sum_{i\in\mathcal{GC}} \max_{\mathbf{p}^i\in\mathbf{P}^i} \left\{ \sum_{n\in\mathcal{N}^i_{\mathcal{GC}}} \Big[\lambda_n p^i_n - c^i_n\Big(p^i_n\Big)\Big] \right\} + \max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left\{ - \sum_{n\in\mathcal{N}_b} \lambda_n f_n(\boldsymbol{\theta}) \right\} \qquad (4.3)
\end{aligned}$$

where the constraint sets are $\mathbf{D}^i = \{\mathbf{d}^i | \underline{\mathbf{d}}^i \leq \mathbf{d}^i \leq \bar{\mathbf{d}}^i\}$, $\mathbf{P}^i = \{\mathbf{p}^i | \underline{\mathbf{p}}^i \leq \mathbf{p}^i \leq \bar{\mathbf{p}}^i\}$, and $\boldsymbol{\Theta} = \{(\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^{T_c}) \in \boldsymbol{\Theta}^1 \times \cdots \times \boldsymbol{\Theta}^{T_c} : \theta^i_n = \theta^j_n, n \in \mathcal{N}^{i,j}_{\mathcal{TC}}, i, j \in \mathcal{TC}\}$ with each TransCo's feasible set defined as

$$\begin{aligned}\Theta^i := \Big\{\boldsymbol{\theta}^i \Big| & g(\theta_{nm}) \le K_{nm}, g\big(\theta_{mn}\big) \le K_{mn},\ \{n,m\} \in \mathcal{E}_l^i; \\ & \underline{\theta}_{nm} \le \theta_{nm} \le \overline{\theta}_{nm}, \{n,m\} \in \mathcal{E}_l^i; \\ & \theta_n = 0, n \in \mathcal{N}_{\mathcal{TC}}^i \cap \mathcal{N}_b^s; \theta_n \in [-\pi,\pi], n \in \mathcal{N}_{\mathcal{TC}}^i\Big\}.\end{aligned}$$

For later reference, the dual problem of Problem (P) is simply

$$\min_{\boldsymbol{\lambda}} \phi(\boldsymbol{\lambda}). \tag{D}$$

4.4.1. Agent Surplus Functions

The arguments of the maximizations in Eq. (4.3) represent surplus functions of the agents. This follows from the fact that the dual variables, $\boldsymbol{\lambda}$, of the power balance equation represent locational marginal prices. The surplus for DistCo $i \in \mathcal{DC}$ for a given demand profile $(\mathbf{d}^i, \mathbf{s}^i)$ at price $\boldsymbol{\lambda}$ is equal to the utility obtained from $\mathbf{d}^i$ minus the cost of total demand (sum of elastic and inelastic demand), defined as

$$\Psi^i_{\mathcal{DC}}(\mathbf{d}^i, \boldsymbol{\lambda}) := \sum_{n \in \mathcal{N}^i_{\mathcal{DC}^d}} \left[u^i_n\left(d^i_n\right) - \lambda_n d^i_n\right] - \sum_{n \in \mathcal{N}^i_{\mathcal{DC}^s}} \lambda_n s^i_n.$$

The surplus of each GenCo $i \in \mathcal{GC}$ is equal to the payment it receives for producing power minus the generation cost,

$$\Psi^i_{\mathcal{GC}}(\mathbf{p}^i, \boldsymbol{\lambda}) := \sum_{n \in \mathcal{N}^i_{\mathcal{GC}}} \left[\lambda_n p^i_n - c^i_n\left(p^i_n\right)\right].$$

TransCos receive a surplus for facilitating power flow across the network. Congestion and losses in transmission lines creates different valuations for power across the network (represented by LMPs) and results in a discrepancy between the payments received from DistCos and the payments made to GenCos. This creates a surplus (possibly negative) for transmitting power from GenCos to DistCos, termed the *merchandizing surplus*. Under the convex DC approximation, the total merchandizing surplus (argument of the last maximization term in Eq. (4.3)) can be shown to be

$$\begin{aligned}\Psi_{\mathcal{TC}}(\boldsymbol{\theta}, \boldsymbol{\lambda}) &= -\sum_{n\in\mathcal{N}_b} \lambda_n f_n(\boldsymbol{\theta}) \\ &= -\sum_{(n,m)\in\vec{\mathcal{E}}_l} \lambda_n g(\theta_{nm}) \\ &= \frac{1}{2}\sum_{(n,m)\in\vec{\mathcal{E}}_l} \Big((\lambda_m - \lambda_n)\bar{g}(\theta_{nm}) - (\lambda_n + \lambda_m)\tilde{g}(\theta_{nm})\Big) \end{aligned} \tag{4.4}$$

where $\vec{\mathcal{E}}_l$ is the directed edge-set and is defined as the set that contains the pair (n, m) and (m, n) for every edge $\{n, m\} \in \mathcal{E}_l$. The quantity $\Big(\lambda_m - \lambda_n\Big)\bar{g}(\theta_{nm}) - \Big(\lambda_n + \lambda_m\Big)\tilde{g}(\theta_{nm})$ is the merchandizing surplus for enabling flow between buses n and m at the price vector $\boldsymbol{\lambda}$. Notice that the first term, $\Big(\lambda_m - \lambda_n\Big)\bar{g}(\theta_{nm})$, is the familiar expression for the merchandizing surplus under the DC approximation [Wu et al., 1996]. The second term, $-\Big(\lambda_n + \lambda_m\Big)\tilde{g}(\theta_{nm})$, arises from the fact that we are considering losses in our model.

Since the ownership of transmission lines is partitioned among TransCos we can separate the total merchandizing surplus into each TransCo's merchandizing surplus as

$$\Psi^i_{\mathcal{TC}}(\boldsymbol{\theta}^i, \boldsymbol{\lambda}) = \frac{1}{2}\sum_{(n,m)\in\vec{\mathcal{E}}^i_l} \Big((\lambda_m - \lambda_n)\bar{g}(\theta_{nm}) - (\lambda_n + \lambda_m)\tilde{g}(\theta_{nm})\Big). \tag{4.5}$$

We show that, via a message exchange process described in Section 4.5.1, TransCos communicate to obtain the angle profile which maximizes $\Psi_{\mathcal{TC}}(\boldsymbol{\theta}, \boldsymbol{\lambda})$ over $\boldsymbol{\theta} \in \Theta$ at price $\boldsymbol{\lambda}$.

4.4.2. Competitive Equilibria in Energy Markets

We can now define the concept of a competitive equilibrium in the context of our energy market model. The definition builds upon the one found in [Motto et al., 2002a].

Definition 4.4.1 (Competitive Equilibrium). *A* competitive equilibrium *is defined as the tuple* $(\{\hat{\mathbf{d}}^i\}_{i\in\mathcal{DC}}, \{\hat{\mathbf{p}}^i\}_{i\in\mathcal{GC}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\lambda}})$ *such that*

(i) $\left\|f_n(\hat{\boldsymbol{\theta}}) - \hat{p}_n + \hat{d}_n + s_n\right\| < \varepsilon$ *for all* $n \in \mathcal{N}_b$, $\varepsilon > 0$

(ii) $\hat{\mathbf{d}}^i(\hat{\boldsymbol{\lambda}})$ *maximizes* $\Psi^i_{\mathcal{DC}}(\mathbf{d}^i, \hat{\boldsymbol{\lambda}})$ *s.t.* $\mathbf{d}^i \in \mathbf{D}^i, \forall\, i \in \mathcal{DC}$

$\hat{\mathbf{p}}^i(\hat{\boldsymbol{\lambda}})$ *maximizes* $\Psi^i_{\mathcal{GC}}(\mathbf{p}^i, \hat{\boldsymbol{\lambda}})$ *s.t.* $\mathbf{p}^i \in \mathbf{P}^i, \forall\, i \in \mathcal{GC}$

$\hat{\boldsymbol{\theta}}^i(\hat{\boldsymbol{\lambda}})$ *maximizes* $\Psi^i_{\mathcal{TC}}(\boldsymbol{\theta}^i, \hat{\boldsymbol{\lambda}})$ *s.t.* $\boldsymbol{\theta}^i \in \Theta^i, \forall\, i \in \mathcal{TC}$

The above definition states that a competitive equilibrium must not only satisfy the power balance equation (condition (i)) but also result in maximum surplus for all DistCos, GenCos, and TransCos (condition (ii)).

4.5. Solution Methodology

Throughout the remainder of the chapter we describe a procedure in which the MO and agents interact in order to obtain a *globally optimal solution* $\mathbf{x}^*$ to the nonconvex primal problem (P). The procedure is based on the dual decomposition method; an iterative method that first involves the evaluation of the dual function for a given set of dual variables (prices), followed by an update of the dual variables.

In the context of the electricity market model in this chapter, the evaluation of the dual function is performed in a distributed fashion by the agents. In fact, maximization of surpluses by the agents corresponds exactly to the evaluation of the dual function. DistCos and GenCos maximize in parallel to obtain the optimal profiles for the current price $\boldsymbol{\lambda}^t$, denoted by $\{\mathbf{d}^i(\boldsymbol{\lambda}^t)\}_{i\in\mathcal{DC}}$ and $\{\mathbf{p}^i(\boldsymbol{\lambda}^t)\}_{i\in\mathcal{GC}}$, respectively. TransCos partake in a message exchange process (due to coupling of merchandizing surplus functions) in order to obtain the operating point which maximizes the total merchandizing surplus at the current price, denoted by $\boldsymbol{\theta}(\boldsymbol{\lambda}^t)$. The MO uses these maximizers to update the price in such a way as to enforce feasibility (condition (i) of Def. 4.4.1). A block diagram outlining the method can be seen in Fig. 4.2.

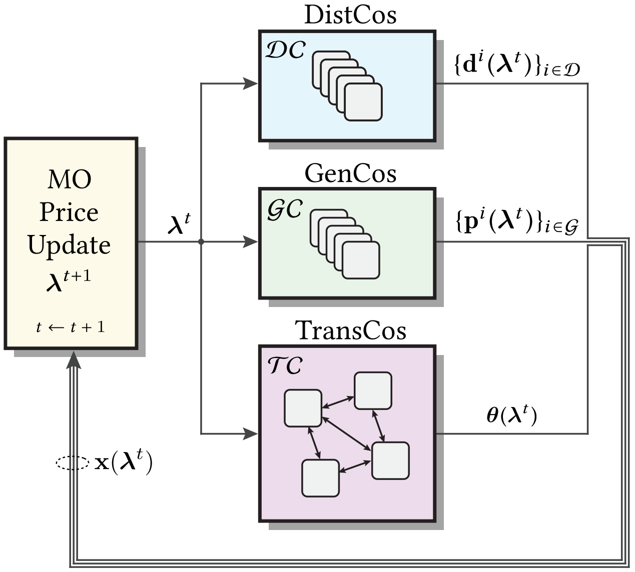

DistCos
$\mathcal{DC}$
$\{\mathbf{d}^i(\boldsymbol{\lambda}^t)\}_{i\in\mathcal{D}}$
MO
Price
Update
$\boldsymbol{\lambda}^{t+1}$
$t \leftarrow t+1$
$\boldsymbol{\lambda}^t$
GenCos
$\mathcal{GC}$
$\{\mathbf{p}^i(\boldsymbol{\lambda}^t)\}_{i\in\mathcal{G}}$
TransCos
$\mathcal{TC}$
$\boldsymbol{\theta}(\boldsymbol{\lambda}^t)$
$\mathbf{x}(\boldsymbol{\lambda}^t)$

Figure 4.2: Outline of the pricing process. Given the current price vector $\boldsymbol{\lambda}^t$, DistCo's and GenCo's update the respective components of the consumption profiles $\{\mathbf{d}^i(\boldsymbol{\lambda}^t)\}_{i\in\mathcal{DC}}$ and generation profile $\{\mathbf{p}^i(\boldsymbol{\lambda}^t)\}_{i\in\mathcal{GC}}$, in parallel. TransCo's participate in a message exchange process to reach an angle profile agreement $\boldsymbol{\theta}(\boldsymbol{\lambda}^t)$. The MO then updates the price to $\boldsymbol{\lambda}^{t+1}$ using the responses $\mathbf{x}(\boldsymbol{\lambda}^t) = (\{\mathbf{d}^i(\boldsymbol{\lambda}^t)\}_{i\in\mathcal{DC}}, \{\mathbf{p}^i(\boldsymbol{\lambda}^t)\}_{i\in\mathcal{GC}}, \boldsymbol{\theta}(\boldsymbol{\lambda}^t))$ (outlined in Section 4.5.2).

4.5.1. Price Response

The first step of the pricing process involves evaluation of the dual function for the current price vector $\boldsymbol{\lambda}^t$. This is achieved via the following agent surplus maximizations.

DistCo Optimizations

Each DistCo, $i \in \mathcal{DC}$, wishes to specify the elastic demand level $\mathbf{d}^i$ in order to maximize its surplus from buying power (both elastic and inelastic) at the current price $\boldsymbol{\lambda}^t$. Each DistCo $i \in \mathcal{DC}$ solves

$$\mathbf{d}^i(\boldsymbol{\lambda}^t) = \operatorname*{argmax}_{\mathbf{d}^i \in \mathbf{D}^i} \Psi^i_{\mathcal{DC}}(\mathbf{d}^i, \boldsymbol{\lambda}^t). \tag{$P^i_{\mathcal{DC}}$}$$

By assumption 2, each u^i_n is strictly concave and therefore the maximizer $\mathbf{d}^i(\boldsymbol{\lambda}^t)$ of Problem ($P^i_{\mathcal{DC}}$) is unique for each i.

GenCo Optimizations

Each GenCo, $i \in \mathcal{GC}$, wishes to specify the injection levels $\mathbf{p}^i$ in order to maximize its surplus from selling power at $\boldsymbol{\lambda}^t$. Each GenCo $i \in \mathcal{GC}$ solves

$$\mathbf{p}^i(\boldsymbol{\lambda}^t) = \operatorname*{argmax}_{\mathbf{p}^i \in \mathbf{P}^i} \Psi^i_{\mathcal{GC}}(\mathbf{p}^i, \boldsymbol{\lambda}^t). \tag{$P^i_{\mathcal{GC}}$}$$

Again, by assumption 2, the maximizer $\mathbf{p}^i(\boldsymbol{\lambda}^t)$ of Problem ($P^i_{\mathcal{GC}}$) is unique for each $i \in \mathcal{GC}$.

TransCo Optimizations

Each TransCo, $i \in \mathcal{TC}$, aims to specify their voltage angle profile $\boldsymbol{\theta}^i \in \Theta^i$ such that the induced flows maximize their merchandizing surplus at the current price, $\Psi^i_{\mathcal{TC}}(\boldsymbol{\theta}^i, \boldsymbol{\lambda}^t)$. Doing so is complicated by the fact that there exist buses that are shared between one or more TransCos, that is, $\mathcal{N}_{\mathcal{TC}}^{i,j} \neq \varnothing$ for neighboring $i, j \in \mathcal{TC}$. The presence of these shared buses creates coupling between the merchandizing surplus functions of distinct TransCos.

As a result, all neighboring TransCos must negotiate the angle value of their shared buses. Arriving at a system-wide agreement for the shared buses, with each TransCo maximizing their own merchandizing surplus, results in a maximization of the total merchandizing surplus (achieving the value of the last term in Eq. (4.3)). The angle profile agreement is achieved via a message exchange process that is based on the ADMM algorithm [Boyd et al., 2011] in which neighboring TransCos iteratively exchange the voltage angle values of their shared buses.

To make use of the ADMM algorithm, it is necessary to write the problem of maximizing the total merchandizing surplus, $\max_{\boldsymbol{\theta}\in\Theta} \Psi_{\mathcal{TC}}(\boldsymbol{\theta}, \boldsymbol{\lambda})$, as the equivalent problem

$$\begin{aligned} \max_{\{\boldsymbol{\theta}^i\}_{i\in\mathcal{TC}}, \mathbf{z}} \quad & \Psi_{\mathcal{TC}}(\boldsymbol{\theta}, \boldsymbol{\lambda}^t) = \sum_{i\in\mathcal{TC}} \Psi_{\mathcal{TC}}^i(\boldsymbol{\theta}^i, \boldsymbol{\lambda}^t) \\ \text{subject to} \quad & \boldsymbol{\theta}^i \in \Theta^i, i \in \mathcal{TC} \\ & \boldsymbol{\theta}^i - \mathbf{z}^i = \mathbf{0}, i \in \mathcal{TC} \end{aligned} \tag{$P_{\mathcal{TC}}$}$$

where $\mathbf{z} \in \mathbb{R}^N$ is the global variable representing the system-wide angle profile $\boldsymbol{\theta}$ and $\mathbf{z}^i = \{z_n\}_{n\in\mathcal{N}_{\mathcal{TC}}^i}$ is the relevant component of $\mathbf{z}$ corresponding to TransCo i's angle profile. We associate a set of dual variables, $\mathbf{y}^i = \{y_n\}_{n\in\mathcal{N}_{\mathcal{TC}}^i}$, with each of the *consensus constraints* $\boldsymbol{\theta}^i - \mathbf{z}^i = \mathbf{0}$, $i \in \mathcal{TC}$. The consensus constraints enforce the angle profiles of TransCos to agree. Defining primal and dual residual norms [Boyd et al., 2011] as

$$p_r^{(k)} := \left(\boldsymbol{\theta}^{1,(k)} - \mathbf{z}^{1,(k)}, \dots, \boldsymbol{\theta}^{T_c,(k)} - \mathbf{z}^{T_c,(k)}\right) \tag{4.6}$$

$$d_r^{(k)} := -\mu\left(\mathbf{z}^{1,(k)} - \mathbf{z}^{1,(k-1)}, \dots, \mathbf{z}^{T_c,(k)} - \mathbf{z}^{T_c,(k-1)}\right) \tag{4.7}$$

the TransCo message exchange process is given by Alg. 2.

Algorithm 2 TransCo Message Exchange Process

```
Initialize $k = 0$, $\mathbf{y}^{i,(0)} = \mathbf{0}$ for all $i \in \mathcal{TC}$, $\mathbf{z}^{(0)} = \mathbf{0}$, $\mu > 0$
while $\neg(||p_r^{(k)}||_2 < \varepsilon_{\text{primal}}$ and $||d_r^{(k)}||_2 < \varepsilon_{\text{dual}})$ do
    for ( do (parallel optimization and broadcast)) $i \in \mathcal{TC}$
        TransCo $i$ solves:
            $$\boldsymbol{\theta}^{i,(k+1)}(\boldsymbol{\lambda}^t) = \underset{\boldsymbol{\theta}^i \in \Theta^i}{\text{argmax}} \left\{ \Psi^i_{\mathcal{TC}}(\boldsymbol{\theta}^i, \boldsymbol{\lambda}^t) - \left(\mathbf{y}^{i,(k)}\right)^{\top}(\boldsymbol{\theta}^i - \mathbf{z}^{i,(k)}) - \frac{\mu}{2} \left\|\boldsymbol{\theta}^i - \mathbf{z}^{i,(k)}\right\|^2 \right\}$$
        Broadcast $\{\theta_n^{i,(k+1)}\}_{n \in \mathcal{N}^{i,j}_{\mathcal{TC}}}$ to neighboring $j \in \mathcal{TC}$;
    end for
    for ( do(parallel average and dual variable update)) $i \in \mathcal{TC}$
        Average: $z_n^{(k+1)} = \frac{1}{|\mathcal{TC}_n|} \sum_{j \in \mathcal{TC}_n} \theta_n^{j,(k+1)}, \forall n \in \mathcal{N}^i_{\mathcal{TC}}$
        Update: $\mathbf{y}^{i,(k+1)} = \mathbf{y}^{i,(k)} + \mu(\boldsymbol{\theta}^{i,(k+1)} - \mathbf{z}^{i,(k+1)})$
    end for
    Update residuals: compute $p_r^{(k+1)}$, $d_r^{(k+1)}$ via Eq.'s (4.6), (4.7)
    Update counter: $k \leftarrow k + 1$
end while
```

By assumption 1, each TransCo i has a slack bus in its set of buses $\mathcal{N}^i_{\mathcal{TC}}$, and the following lemma holds.

Lemma 4.5.1. *The merchandizing surplus function $\Psi^i_{\mathcal{TC}}(\boldsymbol{\theta}^i, \boldsymbol{\lambda})$ is strongly concave in $\boldsymbol{\theta}^i$ for all $i \in \mathcal{TC}$.*

Proof: See Appendix B.1. □

Lemma 4.5.1 and the convergence result of the ADMM in [Boyd et al., 2011] (p.17) lead to the following corollary.

Corollary 4.5.1. *Alg. 1 generates iterates $\{\mathbf{z}^{(k)}\}$ that converge to the unique solution $\boldsymbol{\theta}(\boldsymbol{\lambda}^t)$ of $\max_{\boldsymbol{\theta} \in \Theta} \Psi_{\mathcal{TC}}(\boldsymbol{\theta}, \boldsymbol{\lambda}^t)$.*

All of the maximizers for the current price $\boldsymbol{\lambda}^t$ are then broadcast to the MO, as in Fig. 4.2, and the price is updated.

4.5.2. Price Update

The MO receives the maximizers from the agents for the current price $\boldsymbol{\lambda}^t$, denoted by $\mathbf{x}(\boldsymbol{\lambda}^t) = (\{\mathbf{d}^i(\boldsymbol{\lambda}^t)\}_{i\in\mathcal{DC}}, \{\mathbf{p}^i(\boldsymbol{\lambda}^t)\}_{i\in\mathcal{GC}}, \boldsymbol{\theta}(\boldsymbol{\lambda}^t))$, and uses them to compute an updated price $\boldsymbol{\lambda}^{t+1}$. The price is updated in such a way as to iteratively enforce the power balance equation (see condition (i) of Def. 4.4.1). Before defining the price update, we state the following result.

Lemma 4.5.2. *Under assumption 3, the Hessian of the Lagrangian is negative definite, that is,* $\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}) \prec \mathbf{0}$ *for all* $\mathbf{x}$.

Proof: See Appendix B.2. □

As a consequence of the strong concavity of the Lagrangian, the dual function, Eq. (4.3), is unique and its derivative exists. The gradient of the dual function is (see Thm. 6.3.3 of [Bazaraa et al., 2013])

$$\nabla_{\boldsymbol{\lambda}}\phi(\boldsymbol{\lambda}) = \mathbf{h}(\mathbf{x}(\boldsymbol{\lambda}))^\top \tag{4.8}$$

where $\mathbf{h}(\mathbf{x})$ was defined at the beginning of Section 4.4. As a result, solving the power balance equation is equivalent to finding where the gradient of the dual function vanishes.

The price is updated via a gradient descent algorithm. Specifically,

$$\begin{aligned}\boldsymbol{\lambda}^{t+1} &= \boldsymbol{\lambda}^t - \alpha_t\nabla_{\boldsymbol{\lambda}}\phi(\boldsymbol{\lambda}^t)\\ &= \boldsymbol{\lambda}^t - \alpha_t\mathbf{h}(\mathbf{x}(\boldsymbol{\lambda}^t))\\ &= \boldsymbol{\lambda}^t - \alpha_t\Big(\boldsymbol{f}\left(\boldsymbol{\theta}(\boldsymbol{\lambda}^t)\right) - \mathbf{p}(\boldsymbol{\lambda}^t) + \mathbf{d}(\boldsymbol{\lambda}^t) + \mathbf{s}\Big)\end{aligned} \tag{4.9}$$

with step-size α_t, net generation profile $\mathbf{p}(\boldsymbol{\lambda}^t) = (p_1(\boldsymbol{\lambda}^t), \ldots, p_{n_b}(\boldsymbol{\lambda}^t))$, with net injection $p_n(\boldsymbol{\lambda}^t) = \sum_{i\in\mathcal{GC}} p^i_n(\boldsymbol{\lambda}^t)$, and net elastic demand profile $\mathbf{d}(\boldsymbol{\lambda}^t) = (d_1(\boldsymbol{\lambda}^t), \ldots, d_{n_b}(\boldsymbol{\lambda}^t))$, with net demand $d_n(\boldsymbol{\lambda}^t) = \sum_{i \in \mathcal{DC}} d_n^i(\boldsymbol{\lambda}^t)$. The injection term $\boldsymbol{f}(\boldsymbol{\theta}(\boldsymbol{\lambda}^t))$ is computed from the angle profile $\boldsymbol{\theta}(\boldsymbol{\lambda}^t)$ (from Sec. 4.5.1).

The above recursion is guaranteed to converge to a stationary point of the dual function. To show this, we first demonstrate that the dual function is Lipschitz continuous (this follows from the fact that a function with a bounded derivative is Lipschitz). By Eq. (4.8), the gradient of the dual function satisfies $\nabla_{\boldsymbol{\lambda}} \phi(\boldsymbol{\lambda}) = \mathbf{h}(\mathbf{x}(\boldsymbol{\lambda}))^\top$. Noting that $\|\mathbf{h}(\mathbf{x}(\boldsymbol{\lambda}))\| = \|\boldsymbol{f}(\boldsymbol{\theta}(\boldsymbol{\lambda})) - \mathbf{p}(\boldsymbol{\lambda}) + \mathbf{d}(\boldsymbol{\lambda}) + \mathbf{s}\|$ and the fact that $\boldsymbol{f}(\boldsymbol{\theta}(\boldsymbol{\lambda}))$, $\mathbf{p}(\boldsymbol{\lambda})$, and $\mathbf{d}(\boldsymbol{\lambda})$ are all bounded, there exists some $M < \infty$ such that $\|\nabla_{\boldsymbol{\lambda}} \phi(\boldsymbol{\lambda})\| = \|\mathbf{h}(\mathbf{x}(\boldsymbol{\lambda}))\| \leq M$. Thus the dual function is Lipschitz continuous. Furthermore, notice that the dual function $\phi(\boldsymbol{\lambda})$ is a convex function of $\boldsymbol{\lambda}$. It can be shown through standard arguments that, for a sufficiently small step-size, gradient descent applied to a convex function generates iterates satisfying

$$\phi(\boldsymbol{\lambda}^t) - \phi^* \leq \frac{\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 + \sum_{s=0}^{t} \alpha_s^2 \|\nabla \phi(\boldsymbol{\lambda}^s)\|^2}{2 \sum_{s=0}^{t} \alpha_s}$$

where ϕ^* denotes a minimum of ϕ. In order to ensure convergence, one must choose α_t such that $\sum_{s=0}^{\infty} \alpha_s^2 < \infty$ and $\sum_{s=0}^{\infty} \alpha_s = \infty$. Noting that $\|\nabla \phi(\boldsymbol{\lambda}^s)\| \leq M$ for all s, we have $\phi(\boldsymbol{\lambda}^t) \to \phi^*$. Selecting a step-size of the form $\alpha_t = \beta / t$, $\beta > 0$, ensures that the pricing process converges to a minimizer $\boldsymbol{\lambda}^*$ of the dual function $\phi(\boldsymbol{\lambda})$, solving the dual problem (D).[‡] Since the dual problem is unconstrained, $\nabla \phi(\boldsymbol{\lambda})|_{\boldsymbol{\lambda} = \boldsymbol{\lambda}^*} = \mathbf{0}$, and, again by Eq. (4.8), $\mathbf{h}(\mathbf{x}(\boldsymbol{\lambda}^*)) = \mathbf{0}$.

The convergent dual solution of pricing process results in a zero duality gap with Problem (P) as described by Theorem 4.5.1 below.

Theorem 4.5.1. *The pricing process generates a competitive equilibrium* $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$*, where* $\mathbf{x}^* = \mathbf{x}(\boldsymbol{\lambda}^*)$ *is a globally optimal solution to Problem (P).*

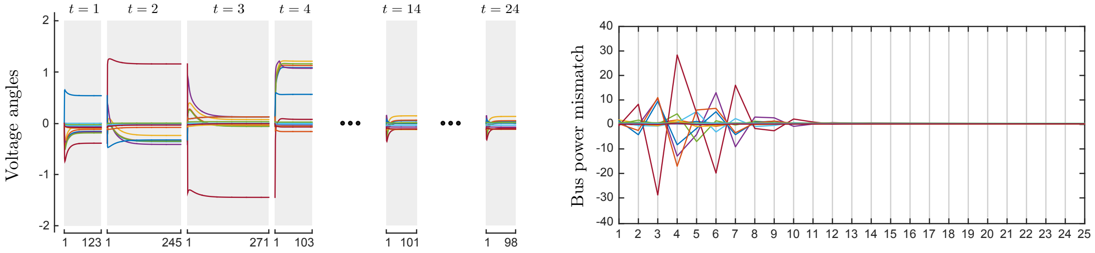
[‡] Prices must satisfy assumption 3 at each iteration t in order to ensure the TransCo subproblems are convex. This can simply be achieved through choice of a sufficiently positive $\boldsymbol{\lambda}^0$.

Proof. See Appendix B.3. □

In summary, the pricing process is guaranteed to generate the competitive equilibrium $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$, resulting in a globally optimal solution $\mathbf{x}^*$ to the (nonconvex) social welfare maximization problem (P). Consequently, $\mathbf{x}^*$ is a Pareto efficient outcome.

4.6. Numerical Example

We demonstrate the performance of the pricing process on a modified version of the IEEE 14 bus test system. The ownership of generators in the modified system is split among three GenCos with $\mathbf{p}^1 = (p_1^1, p_2^1)$, $\mathbf{p}^2 = (p_3^2, p_6^2)$, and $\mathbf{p}^3 = (p_8^3)$. The network also consists of seven DistCos with $\mathbf{d}^1 = (d_2^1, d_3^1)$, $\mathbf{d}^2 = (d_3^2, d_4^2)$, $\mathbf{d}^3 = (d_5^3)$, $\mathbf{d}^4 = (d_6^4, d_{11}^4, d_{12}^4)$, $\mathbf{d}^5 = (d_9^5, d_{10}^5)$, $\mathbf{d}^6 = (d_{12}^6, d_{13}^6)$, $\mathbf{d}^7 = (d_{14}^7)$ and inelastic demands (in MW) $s_2^1 = 15$, $s_5^3 = 10$, $s_{12}^4 = 15$, $s_{10}^5 = 10$, $s_{14}^6 = 15$. The ownership of lines is split among two TransCos, $\mathcal{E}_l^1 = \{\{1,2\},\{1,5\},\{2,3\},\{2,4\},\{2,5\},\{3,4\},\{4,5\},\{4,7\},\{5,6\}\}$, $\mathcal{E}_l^2 = \{\{4,9\},\{6,11\},\{6,12\},\{6,13\},\{7,8\},\{7,9\},\{9,10\},\{9,14\},\{10,11\},\{12,13\},\{13,14\}\}$ with slack bus $\mathcal{N}_b^s = \{6\}$. Parameters for the TransCo message exchange process (Alg. 2) are $\mu = 0.21$, $\varepsilon_{\text{primal}} = 5{\times}10^{-5}$, $\varepsilon_{\text{dual}} = 5{\times}10^{-6}$. Fig. 4.3 demonstrates the convergence of the pricing process.



(a) Algorithm 1 cycle iterations, k

(b) Pricing process iterations, t

Figure 4.3: Convergence of pricing process: **(a)** TransCos reach an angle agreement for each price vector $\boldsymbol{\lambda}^t$ via Alg. 1 (each negotiation cycle corresponds to a price vector); **(b)** The power mismatch at each bus n, $h_n(\mathbf{x}(\boldsymbol{\lambda}^t)) = f_n(\boldsymbol{\theta}(\boldsymbol{\lambda}^t)) - p_n(\boldsymbol{\lambda}^t) + d_n(\boldsymbol{\lambda}^t) + s_n$, converges to zero.

It is evident from Fig. 4.3(b) that the pricing process generates a solution where the power balance equation is satisfied (condition (i) of Def. 4.4.1). At the corresponding prices, agents report their surplus-maximizing responses, satisfying condition (ii) of Def. 4.4.1. By Theorem 4.5.1, the resulting competitive equilibrium is Pareto efficient.

4.7. Discussion and Conclusion

We have presented a mechanism that, through iterative price-response and price-updating, guides the system to a socially optimal outcome. Interestingly, while giving the TransCos the freedom to maximize their merchandizing surplus corresponds to them attempting to congest their lines (since a larger power flow results in a higher merchandizing surplus), this behavior is required for ensuring convergence to an efficient outcome. Furthermore, it is important to note that giving the TransCos this freedom does not necessarily mean that the resulting operating point will result in congested lines.

In summary, this chapter discussed the development of an electricity market model and an associated decentralized market mechanism that, under natural assumptions, ensures convergence to a Pareto efficient market (competitive) equilibrium. The market model includes multiple DistCos, GenCos, and (cooperative) TransCos all of which are assumed to be surplus-maximizing given the current set of LMPs. A market operator updates LMPs via a gradient method in order to achieve an operating point that satisfies the power balance equations and consequently clears the market.

Part II

Dynamic Security of Cyber-Physical Systems under Partial Information

Chapter 5

Cyber-Physical Systems Security

It won't be difficult for society to adjust to the conveniences brought on by cyber-physical systems. Unfortunately, our high reliance upon these systems, combined with their widespread integration into nearly every aspect of our lives, will make us very sensitive to their failures.

Recent events have demonstrated the scale of the disruption when cyber-physical systems fail, especially those related to critical infrastructure. A prime example is the blackout of 2003 that spanned the midwest and northeast regions of the United States as well as parts of Canada [Abraham & Efford, 2004]. The failure was triggered by a sagging transmission line coming into contact with foliage, causing it to trip and go offline. Due to a malfunction in an alarm notification system (resulting from a software bug in General Electric's XA/21™ energy management system [Poulsen, 2004]), the loss of the transmission line went unnoticed. Making matters worse, the region's state estimation system was not fully functional (due to human error), resulting in an incomplete view of the system's current operating status. These issues resulted in a sequence of cascading failures that operators were unable to recognize in time to resolve. When the cascade eventually came to an end, 508 generators were offline leaving more than 50 million people without power.

Another example of a wide-spread failure event is the power outage that impacted Delta airlines in 2016. Due to a malfunction in a power control module at Delta's headquarters in Atlanta, a transformer overloaded and took many of the airline's servers offline. According to an interview with Delta's CEO Ed Bastian [Yamanouchi, 2016], "300 of [Delta] airline's 7000 servers were not wired to backup power." This oversight caused a system-wide crash, resulting in more than 2000 flight cancellations world-wide, displacing large numbers of customers, crew, and aircraft.

The above examples highlight the disruptions that critical infrastructure failures can have on society. Fortunately, the likelihood that a catastrophic sequence of (essentially random) events will occur is quite low and helps to explain why events of this scale are relatively rare. That said, a particularly concerning realization is the wide-spread damage that could result if events were triggered by an agent with malicious intent. While rare, we have already started to see such intelligent, targeted attacks on critical infrastructure systems. Due to our increased reliance on these systems, attacks of this nature have the potential to significantly disrupt our everyday life, necessitating the study of how they unfold and the design of defense systems that prevent them from succeeding.

5.1. An Emerging Class of Attacks

Attacks on cyber-physical systems have started to emerge that exploit the deep connectivity of the cyber layer with the physical infrastructure. The ability to control a physical process from the cyber infrastructure, coupled with the growing connectivity of our societal systems, has introduced multiple attack pathways for malicious agents, allowing them to influence and potentially permanently damage the physical infrastructure. The two case-studies described below, Stuxnet and the Ukrainian power grid attack, illustrate the complexity of such attacks.

5.1.1. Stuxnet

Stuxnet is one of the most sophisticated attacks ever seen. First detected in 2010, the attack was targeted at Siemens programmable logic controllers (PLCs) with the intention of interfering with centrifuges at Iran's Natanz fuel enrichment plant [Cherry, 2010, Albright et al., 2010, Falliere et al., 2011]. The complexity of Stuxnet was unprecedented, involving extensive use of insider information and many stages of exploits. The attack evolved in four steps: spread, discovery of target computers, disruption of physical processes, and evasion of detection [Falliere et al., 2011]. Stuxnet spread through the local network using combinations of both exploits (including zero-days) and infected removable drives. The use of removable drives allowed Stuxnet to cross the airgap and reach computers capable of (re)programming the PLCs responsible for centrifuge control [Falliere et al., 2011]. Before injecting malicious code into the PLC, Stuxnet measured the operation of the controller for a period of time, checking if a "specific program [was] running on the PLC" [Cherry, 2010], in turn allowing it to conclude that the PLC was indeed controlling a centrifuge. At this point, Stuxnet injected malicious code to modify the frequency set-points of the centrifuge rotors. In order to evade detection, Stuxnet fed back (the previously measured) normal behavior to the monitoring systems, fooling operators and evading automated anomaly detection systems. Furthermore, it also made use of stolen authenticity certificates to evade antivirus software. According to a 2010 report published by the Institute for Science and International Security [Albright et al., 2010], "It is increasingly accepted that, in late 2009 or early 2010, Stuxnet destroyed about 1,000 IR-1 centrifuges out of about 9,000 deployed at the site." Stuxnet represents the first case where physical infrastructure was damaged by malicious code.

5.1.2. Ukrainian Power Grid Attack

In December of 2015, the computer systems controlling the western region of the Ukrainian power grid were hacked. Investigations revealed that the attack was initiated nearly a year prior, with hackers carrying out spear-phishing attacks (malicious emails) on the workers' computers [Zetter, 2016]. These attacks involved the use of malware, termed BlackEnergy3, which served to open a backdoor on the substation's systems [Assante, 2016, Pultarova, 2016]. Using this backdoor, the hackers spent the next few months performing reconnaissance, obtaining worker's VPN credentials and permitting remote access to the system. The hackers used these credentials to modify critical elements of the system, such as corrupting the uninterruptible power supply, resulting in a loss of back-up power to the control centers, and injecting malware, termed KillDisk, preventing workers from being able to remotely control the system [Pultarova, 2016, Assante, 2016, Zetter, 2016]. When the attack was launched on December 23, 2015, the hackers were able to remotely open multiple substations' breakers, disconnecting them from the grid and cutting the power to large regions of the country, all while the workers were unable to do much to stop it. Furthermore, hackers launched a denial-of-service attack on the phone systems, preventing customers from being able to report outages [Pultarova, 2016]. While the attack did not do any permanent damage to physical infrastructure, it did disconnect 230000 people from the grid. The event represents the first time that an attack on critical infrastructure has impacted a civilian population [Cherepanov & Lipovsky, 2016].

5.2. Key Features in Cyber-Physical Systems Security

Analyzing the nature of failure events, of both non-malicious and malicious origin, is helpful for identifying the key features that should be considered when designing secure cyber-physical systems.

(1) **Successful attacks involve multiple levels of exploits across numerous attack vectors**. In the above attack examples, a chain of multiple exploits needs to be successful in order for the attacker to fulfill its objective. These exploits take advantage of vulnerabilities across multiple system components, giving the attacker access to various attack pathways into the system. Thorough analysis of the Stuxnet attack [Langner, 2013] revealed that there were two distinct attack vectors that could have resulted in centrifuge damage: the rotor overspeed attack outlined in Section 5.1.1, as well as a more advanced overpressure attack. As systems grow in complexity, one must reason about a large number of possible attack vectors in the system in order to possess an accurate view of its security and to guide appropriate defense decisions.

(2) **Defense decisions must be made in real-time and subject to partial/noisy information**. Attackers take extensive measures to remain stealthy and evade detection, resulting in one having only partial information of their current capabilities and strategy/intent. Defense decisions must be made in the presence of this uncertainty. Specifically, defense systems must be able to efficiently translate the information provided by noisy security alerts (subject to both missed detections and false alarms) into defense decisions. Efficient processing of security alerts is especially important in the context of cyber-physical systems, where myopically reacting to false alarms could have catastrophic consequences on the availability of the underlying system, *e.g.* interfering with the operation of a flight control system.

(3) **The severity of the attack and subsequent failure depends on the status of the underlying physical system**. One of the factors that contributed to the large scale of the 2003 blackout was the fact that the system was stressed at the time of the initial trigger event. In principle, an intelligent attacker could maliciously trigger a physical failure such that the resulting cascade does maximal damage to the system, *e.g.* by opening a breaker on a heavily-loaded transmission line. Maintaining the security of the system should thus involve reasoning about the attacker's capabilities in the context of the current operating status of the physical system.

(4) **A given attack can unfold on a wide range of time-scales**. Investigations into both Stuxnet and the Ukrainian power grid attack revealed that the attacks took place over many days and months. Stuxnet was programmed to be patient, lying dormant

for up to 27 days between successive frequency modifications [Falliere et al., 2011]. This had the effect of reducing its visibility to system operators and automated detection systems. In the Ukrainian power grid attack, hackers stole workers' credentials which allowed them to perform reconnaissance and corrupt system components many months before the power outage attack occurred. At the other end of the spectrum, we've witnessed attacks that unfold very quickly, as was the case with the WannaCry malware attack [Security Response Team, 2017]. As a result, in order to accurately infer the capabilities and intent of the attacker, one must be able to piece together evidence from drastically different time-scales.

(5) **The target system may not recognize becoming infected or suffering a loss of control**. A particularly concerning feature of sophisticated attacks is that the attacker and/or malicious code can spread among a large number of hosts without being detected. Furthermore, the malicious code can modify the operation of the physical system without the target system recognizing this loss of control. For example, as of September 29, 2010, Stuxnet was present among approximately 100000 hosts spanning many countries [Falliere et al., 2011]. Once on the target machines, Stuxnet was able to modify their operation without the operators' knowledge. This raises concerns that sophisticated malware could spread among many millions of devices around the world, either modifying their operation covertly or lying dormant and waiting for a trigger event, all without our knowledge.

5.3. Overview of Part II

Developing models that are able to capture all of the above features is a difficult task. The objective of the model developed in Chapter 6 is to describe what pathways an attacker can take to infiltrate a system (feature (**1**)) while enabling real-time threat assessment and response selection subject to uncertainty (feature (**2**)). The model is built upon the notion of an attack graph, which serves to describe the causal dependencies between security conditions (attacker capabilities) and exploits. The graph allows one to model the dynamics of the attacker, *i.e.* how the attacker may use its capabilities to perform exploits and gain further capabilities, and provides a basis for quantifying the system's security (via a security state). Through consideration of multiple attacker types, the model is able to capture a wide range of attacker strategies (behavior). Using noisy security alerts, generated as the attacker progresses through the network, the defender constructs a belief over the attacker's capabilities and true strategy. The belief provides context for efficiently processing subsequent security alerts, especially in settings where the false-alarm rate is high. A sampling-based algorithm allows for online prescription of effective defense actions. A discussion of how the proposed model is useful for addressing feature (3), in the context of the electrical grid, is also presented. Features (4) and (5) are not considered in this work.

The requirement to make decisions over time under imperfect information (sequential decision-making under uncertainty) is fundamental to problems related to security. Obtaining the solution to these problems, *i.e.* determining an optimal policy, poses significant theoretical and computational challenges. Chapter 7 investigates conditions under which a specific class of sequential-decision problems, POMDPs, possess optimal policies that are monotone in the belief. Motivated by the model of Chapter 6, specifically the fact that we cannot always say whether one security state is *safer* than another, the model of Chapter 7 studies settings where the underlying state space is partially ordered. The partial ordering of the state space requires the development of a new stochastic order. This stochastic order has many desirable properties, allowing one to establish monotonicity properties of the value functions and dynamic programming recursion, and resulting in monotone optimal policies in a two-action setting.

Chapter 6

A POMDP Approach to the Dynamic Defense of Large-Scale Cyber-Physical Systems

6.1. Introduction

The high connectivity of modern cyber networks and devices has brought with it many improvements to the functionality and efficiency of our networked systems. Unfortunately, these benefits have come with the introduction of many new entry points for attackers, making our systems much more vulnerable to intrusions. Recent events, such as information leakage and theft [Finkle & Skariachan, 2013], car hacking [Greenberg, 2015], and denial-of-service attacks [Etherington & Conger, 2016], have highlighted this vulnerability. Particularly concerning is that the operation of critical infrastructure is becoming increasingly reliant upon (potentially insecure) networked systems, generating significant vulnerabilities in many areas of society. As reported by the Department of Homeland Security's Industrial Control Systems Cyber Emergency Response Team (ICS-CERT), attacks on critical infrastructure sectors (such as manufacturing, energy, communication, water, and transportation systems) have remained persistent over the past few years, with 245 in 2014, 295 in 2015, and 290 in 2016 [Department of Homeland Security, 2016]. Unfortunately, due to the increased reliance of these systems on cyber networks, coupled with an escalation in the sophistication of cyber attacks, many of the recent intrusions have had the potential to inflict severe and widespread damage (an increasing number of attacks have reached the *control system layer* of the system [Department of Homeland Security, 2016].) It is imperative that methods are developed to detect and mitigate these attacks in order to ensure the secure operation of society's critical systems.

One approach to mitigating attacks is, upon discovery of a vulnerability, to develop and release a patch to remove the vulnerability. Unfortunately, the period between discovery of a vulnerability and the application of a patch (termed the *vulnerability exposure window*) is long, often lasting on the order of five months or more [Gorenc & Sands, 2017]. This significant delay results in many cyber networks being operational while multiple known vulnerabilities are present, resulting in significant risks to society. This concern necessitates the development of an active defense system that is capable of taking into account information in real-time, inferring the security status of the system, and translating this information into appropriate defense decisions that are able to immediately respond to and mitigate the progression of the attacker through the system.

The development of such a defense system is complicated by the fact that sophisticated and targeted cyber attacks, especially those carried out by nation-states, rarely consist solely of an exploitation of a single vulnerability. Rather, these attacks usually consist of a complex sequence of exploits, combining many vulnerabilities across multiple system elements, enabling the intruder to infiltrate deep within the cyber network. In an attempt to address these concerns, researchers in the security community have developed theoretical tools (predominantly graphical approaches) to model the complex interactions between vulnerabilities. Attack trees/graphs are a popular formalism for modeling such interactions. First introduced by [Schneier, 1999], attack trees model the dependencies between exploits and *system states*[*] in a cyber network, allowing one to construct the specific attack paths that intruders can take to enter a network. Unfortunately, attack trees and graphs can be enormously large even for modestly-sized systems [Sheyner et al., 2002], restricting their applicability to realistically-sized cyber networks. In order to improve scalability, researchers proposed an assumption on the attacker's behavior, termed *monotonicity* [Ammann et al., 2002], which states that the success of a previous exploit will not interfere with the success of a future exploit. Monotonicity enables one to restrict attention to dependencies between exploits and *security conditions* (system attributes), in what is termed a *dependency graph*, avoiding the need to enumerate over all system states. This enables a more compact representation, allowing one to significantly reduce the amount of information required to describe attacks.

Knowledge of how an attacker can infiltrate a system offers a useful starting point for defining appropriate defenses; however, efficiently processing the available information and translating it into the prescription of an effective defense decision is still a difficult task. One difficulty arises from how to quantify the security status of the system at any given time. The security status is constantly changing as a function of both the attacker's progression through the system and the defender's actions. Furthermore, the defender does not know the true strategy of the attacker and is unable to perfectly observe the attacker's actions, resulting in a lack of certainty of the security status of the system at any given time. The defender only has access to a stream of noisy security information generated in real-time

[*] System states represent an assignment of values to *system attributes* such as: active services (and the associated vulnerabilities), network connectivity, trust relationships between hosts, and attacker privileges on hosts [Sheyner et al., 2002].

(for example, security alerts generated via intrusion detection systems). Oftentimes, this information suffers from a high-rate of false alarms, that is, alarms being triggered when nothing of concern has actually occurred. Furthermore, the defender's choice of a defense action is complicated by its uncertain effects on the security status of the system (due to the defender's uncertainty regarding the true security state) as well as the need to strike a trade-off between enforcing security and maintaining the availability of network resources to trusted users.

In this chapter, we propose a formal model, based on the theory of stochastic control, for selecting defense actions in real-time in order to mitigate the progression of an attacker through the system while minimizing the negative impact to availability. We use a condition dependency graph to model how the attacker progresses through the cyber network over time. We represent the dependency graph as a *hypergraph*, where nodes represent possible security conditions and directed *hyperedges* (edges that connect a pair of sets of nodes) represent exploits, relating preconditions, the security conditions that must be true in order for the exploit to be attempted, to postconditions, the security conditions that become true if the exploit is successfully carried out. Each security condition can either be enabled or disabled, where an enabled condition is interpreted as the attacker possessing a particular capability. We define a *security state* to be the set of currently enabled security conditions. In this sense, the security state at any given time represents the current capabilities of the attacker. For a given security state, the attacker uses its current capabilities (the set of enabled security conditions) to attempt exploits, with the goal of reaching one or more *goal conditions*. The specific strategy that the attacker employs is its own private information and is assumed to dynamically adjust according to the deployed defense decision. In order to model the defender's uncertainty of the attacker's strategy, we consider the attacker to be one of a finite set of *attacker types*. Consideration of many types allows one to capture a wide-range of potential attacker behavior. Each type characterizes the nature of both the security state dynamics (how the attacker progresses through the system, via *probabilities of attack and success* for each exploit) as well as the observation dynamics (the nature of how the intrusion detection system generates security alerts as a function of the attacker's progression, via *probabilities of detection* for exploit attempts and *probabilities of false alarm* for alerts). The defender is able to interfere with the progression of the attacker by performing system modifications that have the effect of blocking exploits from succeeding. The defender possesses uncertainty over both the current capabilities and the true strategy of the attacker and must make its defense decisions based on its *belief matrix*, that is, the joint distribution over security states and attacker types. This belief, constructed such that it is consistent with the defender's available information (the history of security alerts and previously deployed defense actions), summarizes all of the necessary information for making an optimal decision. Through appropriate assignment of costs to both security states and defense actions, we are able to quantify the tradeoff between maintaining security and preserving availability of the system. The resulting defense problem is a *partially observable Markov decision process* (POMDP), the solution of which is a *defense policy* that maps the current belief (of the security state and attacker strategy) to a defense action.

Due to the high dimensionality of the defense problem, scalability of the solution approach is a primary concern. We employ an online algorithm, based on the *partially observable Monte-Carlo planning* (POMCP) algorithm [Silver & Veness, 2010], that simulates future possible state trajectories from the current belief in order to evaluate the effectiveness of various defense decisions, enabling the defender to make a selection in real-time. While forming the basis for our algorithm, the standard POMCP algorithm is not directly suitable for application to our problem. In particular, the belief update procedure does not scale to large observation spaces. As a result, using the context provided by the (belief over the) security state, we take advantage of the structure of the observation process in order to design an efficient belief update procedure that effectively scales to high-dimensional settings. The proposed online defense algorithm enables us to compute good quality defense policies for large instances of the defense problem, overcoming an important obstacle to deployment in realistic cyber network settings.

6.1.1. Literature Review

Systems that select defense actions in response to security alerts are referred to as *intrusion response systems* (IRSs) in the cybersecurity literature. Early IRSs took the form of *passive* systems, logging security information and notifying human operators in order for manual response actions to be selected. Unfortunately, this process is slow and has proven to be inadequate for defending networks against sophisticated modern-day attacks.[†] Consequently, researchers have turned to the development of *active* systems that are capable of automatically responding to intrusions without the need for a human operator to intervene. Such systems are referred to as automated IRSs in the literature.

The past two decades have seen an increasing amount of research in automated IRSs. For literature reviews of the area, the reader is directed to the surveys by [Foo et al., 2008], [Shameli-Sendi et al., 2012], and [Inayat et al., 2016]. Automated IRSs can largely be categorized into two groups: static and dynamic. Static IRSs focus on designing an attack-response map that is capable of executing preprogrammed responses upon detection of attacks (see for example, the work by [Ryutov et al., 2003]). Static approaches, as the name suggests, use a fixed mapping (look-up table) from detected attacks to responses, and consequently, as stated by [Lewandowski et al., 2001], select responses that can be potentially predicted and exploited by an attacker. Furthermore, static IRSs do not take into account the potentially negative side-effects of deploying defense actions and can thus unintentionally inflict further damage to the system. Due to these concerns, researchers began to develop *dy-*

[†] As stated by [Balepin et al., 2003], “some of the most intense intrusions are automated.”

namic IRSs. Dynamic IRSs are capable of factoring in additional information, such as the effectiveness of previously deployed defense actions, *e.g.* [Ragsdale et al., 2000, Foo et al., 2005], or the cost of defenses, *e.g.* [Lee et al., 2002, Toth & Kruegel, 2002, Kheir et al., 2010], in order prescribe a situation-dependent response to mitigate the attack. The ability of dynamic IRSs to modify their response based on new intrusion information raises the bar for the adversary, proving to be much more difficult to circumvent than static IRSs.

One class of dynamic IRSs, termed *state-based* approaches, has received an increasing amount of attention in recent years [Lewandowski et al., 2001, Kreidl & Frazier, 2004, Zonouz et al., 2014, Miehling et al., 2015, Iannucci et al., 2016, Iannucci & Abdelwahed, 2016]. State-based IRSs aim to quantify the security status of a network via the assignment of a security state and enable one to study how this state evolves as a function of both the attacker's and defender's actions. As argued by [Iannucci et al., 2016, Iannucci & Abdelwahed, 2016], a state-based approach allows one to cast the problem of designing an automated IRS as a problem of choosing defense actions that ensure the security state remains in a desirable region of the state space. State-based approaches also allow one to avoid the issue of crafting individual response actions for each attack, since a single defense action may modify the dynamics of the security state's evolution in such a way as to prevent many attacks from being successfully carried out. One of the first to develop a state-based IRS was [Lewandowski et al., 2001]. The authors proposed a state-based approach in order to enable "global situational awareness," ensuring that the selection of a defense action benefits the entire system and not just a localized region. While significant in its contribution, the approach taken in [Lewandowski et al., 2001] does not leverage any formal theory.

The nature of state-based IRSs make them a good fit for the application of formal tools. A well-designed IRS must be able to quickly select defense actions over time when provided with noisy security alert information (including false negatives and false positives) and evaluate the effectiveness of previous defense decisions, all while balancing inherent tradeoffs in the system, such as the conflicting objectives of security and availability. The tools found in control and game theory are well-suited for addressing these requirements, a fact that has been recognized by some in the security community, [Kreidl & Frazier, 2004, Zonouz et al., 2014, Miehling et al., 2015, Iannucci et al., 2016, Iannucci & Abdelwahed, 2016]. One of the first to apply formal theory, namely stochastic control theory, to the design of an automated IRS was [Kreidl & Frazier, 2004] in the development of their system αLADS (ALPHATECH Lightweight Autonomic Defense System). The authors proposed a host-based IRS that receives alerts (as inputs) from an anomaly sensor in order to calculate the probability that the host is in an attack state. The approach uses a POMDP to select countermeasures in order to interfere with the progression of the attacker while attempting to minimize the negative impact to the normal operation of the system. [Zonouz et al., 2014] formulate an automated, network-based IRS as a two-player, sequential Stackelberg stochastic game, termed the Response and Recovery Engine (RRE). The proposed scheme decomposes the problem into a hierarchical structure of local engines (hosts) and a global engine. Local engines contain graphs, termed *attack-response trees* (ARTs), that serve to quantify the security of the hosts based on noisy security alerts. The security information of each host is sent to the global engine which is responsible for computing defense actions. The defense actions are chosen using a (heuristic) fuzzy logic control-based technique under the behavioral assumption that the attacker will attempt to inflict maximum damage to the system. In previous work, [Miehling et al., 2015], we developed a defense scheme that used Bayesian attack graphs (see [Liu & Man, 2005] for the definition) to model the progression of the attacker and quantify the security state. Using noisy security alert information, the defender maintains a belief over the current progression of the attacker. The resulting problem of choosing defense actions over time as a function of the belief is cast as a POMDP. More recently, [Iannucci et al., 2016, Iannucci & Abdelwahed, 2016] proposed an autonomic IRS that uses a Markov Decision Process (MDP) to specify a sequence of defense actions to drive the system back to a normal operating state. They also offer a performance evaluation of their proposed solution method.

The IRS proposed in this chapter differs from existing state-based approaches in multiple ways. First, in the host-based IRS developed by [Kreidl & Frazier, 2004], only the state of the host is taken into consideration when determining the security status of the system. In our model, embedding a state space on the dependency graph allows for the security of the entire network to be taken into account. Furthermore, due to the coarse-grained, small state space in [Kreidl & Frazier, 2004], the scalability problem is not addressed. Second, while the network-based IRS introduced by [Zonouz et al., 2014] addresses the scalability problem via a hierarchical decomposition, our model presents an alternate approach that addresses scalability by employing a Monte-Carlo sampling approach. Additionally, our model uses an expected cost criterion, a less conservative objective than the worst-case cost found in [Zonouz et al., 2014]. Third, compared to our previous work, our current model is more expressive than the model we proposed in [Miehling et al., 2015], allowing for one to consider more complex dependencies between exploits (the model allows for exploits that have multiple postconditions), a more realistic observation model (alerts are triggered by exploit activity and are subject to false alarms), and private attacker strategies. Furthermore, we directly address the scalability concerns in this chapter. Lastly, while [Iannucci et al., 2016, Iannucci & Abdelwahed, 2016] address the state space explosion problem, their work assumes complete observability of the underlying state, whereas our model allows for imperfect observations.

6.1.2. Contribution

The formalism in this chapter offers a quantitative model for the computation and analysis of defense policies under a wide-range of attacker strategies. The specific contributions are as follows:

1) ***Quantification of security***: The model of this chapter is the first to embed a state space on a dependency graph for the purposes of designing a dynamic IRS. Such an approach allows one to accurately quantify the progression of the attacker along (a combinatorial number of) attack pathways, and provides valuable information for selecting defense actions that optimally mitigate the attacker's progression while minimizing the impact to availability. Furthermore, allowing the defender to possess uncertainty over the true underlying (dynamic) attack strategy leads to a more realistic model of attacker-defender interactions, permitting a more accurate quantification of the system's security status.

2) ***Management of false alarms***: The security state provides context for which exploits the attacker has already performed, and which exploits it needs to carry out in order to achieve its goals. Such information is valuable for efficiently processing security alerts, allowing the defender to weigh new security alert information by the likelihood of states in the current belief. That is, the belief is informative for assessing probabilistically whether the given alerts were generated by valid exploit attempts or were simply false alarms. This feature of our model, described in more detail in Section 6.3.3, is particularly useful in settings where there is a high-rate of false alarms, a characteristic of many modern IDSs.

3) ***Scalability***: Even though the number of security states can be very large for some instances of our model, the online defense algorithm (discussed in Section 6.3.1) does not require one to construct the entire state space. Instead, the algorithm samples regions of the state space relevant to the current defense decision, allowing one to avoid the state space explosion problem. This feature, combined with some problem-specific modifications (taking advantage of the structure of the observation process) allows for computation of defense policies in realistically sized domains.

6.2. The Dynamic Security Model

The proposed dynamic security model provides a formal basis for how a defender can detect and mitigate the infiltration of an attacker in a cyber network. Throughout the description of the model, the diagram of Fig. 6.1 will be useful. In particular, the remainder of Section 7.2 will describe the model for the attacker's progression through the cyber network (Section 6.2.1), the defender's quantification of this progression via a security state (Section 6.2.2), the evolution of the security state as a function of the interactions between the attacker and defender (Section 6.2.3), the defender's information and its formation of consistent beliefs (Section 6.2.4), and finally the formulation of the defender's problem (Section 6.2.6).

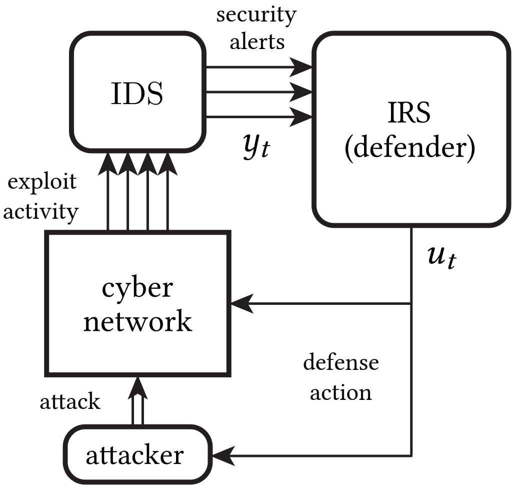


Figure 6.1: The dynamic security model. The attacker progresses through the cyber network by performing exploits, triggering security alerts via an intrusion detection system. The defender uses this intrusion information to construct a belief of the attacker's capabilities and strategy, which is then used to prescribe a defense action.

6.2.1. The Condition Dependency Graph

Researchers and cybersecurity analysts have long been interested in how to represent the steps that intruders take when compromising a system. The concept of attack trees and graphs were developed with this goal in mind, allowing one to study all possible sequences of exploits that an intruder can take to infiltrate a network and reach its goal(s). An attack graph consists of *system states* (nodes) and *transition relations* (edges), which relate system states to each other via exploits. The construction of an attack graph requires one to enumerate over all system states, a process which generates graphs that quickly grow in dimension.

Making assumptions regarding the attacker's behavior allows us to greatly simplify attack graphs and reduce the amount of information required to describe an attack. One such assumption, termed monotonicity [Ammann et al., 2002], states that the success of an exploit does not render the precondition of any other exploit invalid. In simpler terms, the success of one exploit does not interfere with the attacker's ability to carry out a future exploit.[‡] Under monotonicity, one does not need to enumerate all system states in an attack graph, but can rather construct a *dependency graph* describing how exploits relate to *security conditions* [Ammann et al., 2002, Noel & Jajodia, 2004]. The appeal of the dependency graph representation is that the graph can more easily be constructed for large networks, proving to be especially useful in cases where the corresponding attack graph would be intractably large to generate. In the approach taken by [Ammann et al., 2002], the authors construct such a graph where nodes represent security conditions and edges represent exploits in what is termed a *condition dependency graph*.[§] Security conditions are atomic facts (they can either be true or false) that can reflect any of the aforementioned

[‡] See Section 2 of [Ammann et al., 2002] for an explanation of how the majority of non-monotonic attacks can be modeled as monotonic under reasonable assumptions on the attacker's behavior.

[§] This graph has a dual representation termed an *exploit dependency graph* [Noel & Jajodia, 2004, Jajodia et al., 2005].

system attributes.[¶] Exploits relate security conditions via *preconditions* and *postconditions*.

We adopt an approach similar to that of [Ammann et al., 2002] for modeling attack pathways, using a condition dependency graph to represent the dependencies between security conditions and exploits. As discussed by [Ammann et al., 2002], the edges in a condition dependency graph relate the security conditions "in a complex way," where a given exploit can have "both multiple preconditions and multiple postconditions." We formalize this notion by recognizing that such edges are in fact directed *hyperedges* (an "edge" that connects two *sets* of nodes rather than simply a pair of nodes). For simplicity, we adopt a slightly modified definition for the security conditions from the one found in [Ammann et al., 2002]. The security conditions in [Ammann et al., 2002] represent a mix of attributes that are true under the normal network configuration (termed *initial conditions*, such as default network connectivity and active services) and attributes that can be maliciously made true during an attack (which we term *attack conditions*, such as attacker privileges or unintended trust relationships between hosts). We do not include the conditions representing the normal network configuration (the initial conditions) explicitly in the dependency graph, but instead assume that the set of security conditions consists solely of attack conditions. This modification is purely for convenience; under the modified definition, the condition dependency graph for a network that has not yet been subject to an attack has all of its conditions set to false.

Formally, we represent a condition dependency graph as a directed acyclic hypergraph $\mathcal{H} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{c_1, \ldots, c_{n_c}\}$ is the set of security conditions (nodes) and $\mathcal{E} = \{e_1, \ldots, e_{n_e}\}$ is the set of exploits (hyperedges). The acyclic nature of the graph follows from the monotonicity assumption. As discussed earlier, each security condition $c_i \in \mathcal{N}$ in the hypergraph can either be true or false. The truth value of each condition is interpreted

[¶] The important distinction between a system state in an attack graph and a security condition in a dependency graph is that, in attack graphs, each node represents a state, where each state is an assignment of values for *all* of the attributes, whereas in the condition dependency graph, each node represents a single attribute.

as follows: a true (*enabled*) condition means that the attacker possesses condition c_i, and a false (*disabled*) condition means that the attacker does not possess c_i, where an enabled condition is interpreted as the attacker having a particular capability. For example, an enabled condition could mean that the attacker has maliciously enabled a trust relationship between two hosts or has user access on a specific host (where a different privilege level on the same machine would be represented by a distinct condition). Some of the conditions in the hypergraph, when enabled, designate that an attacker has reached a goal. Such nodes are termed *goal conditions* and are denoted by the subset $\mathcal{N}^g \subseteq \mathcal{N}$. Goal conditions are defined by the defender and correspond to something that it wants to protect. For example, a goal condition could represent the attacker possessing root access on a critical host or access to a server that contains sensitive information. It is assumed that the attacker is attempting to enable one of these goal conditions; however, we (as the defender) do not know which one(s). Each hyperedge $e_i \in \mathcal{E}$ represents an exploit and takes the form of an ordered pair of sets, $e_i = (\mathcal{N}_i^-, \mathcal{N}_i^+)$, where $\mathcal{N}_i^- \subseteq \mathcal{N}$ represents e_i's *preconditions* and $\mathcal{N}_i^+ \subseteq \mathcal{N}$ represents e_i's *postconditions*. It is assumed that the attacker is able to attempt exploit e_i only if all preconditions $j \in \mathcal{N}_i^-$ are enabled. This is without loss of generality since for cases where multiple sets of conditions allow for an exploit to be attempted (a disjunction over preconditions), we simply duplicate the exploit for each of its sufficient sets of preconditions. There exist some exploits $e_i \in \mathcal{E}$ with $\mathcal{N}_i^- = \varnothing$, that is, an exploit with an empty set of preconditions. These exploits, termed *initial exploits* and denoted by $\mathcal{E}_0$, represent entry points for the attacker and reflect the fact that they can be performed without the attacker needing any prior (maliciously enabled) capabilities. If an attempted exploit is successful, all postconditions $j \in \mathcal{N}_i^+$ become enabled, increasing the attacker's set of capabilities, allowing it to perform additional exploits and penetrate further into the system. A pictorial representation of a condition dependency graph is provided in Fig. 6.2. We will use this example graph throughout the chapter to aid in the explanation of the model and the results.

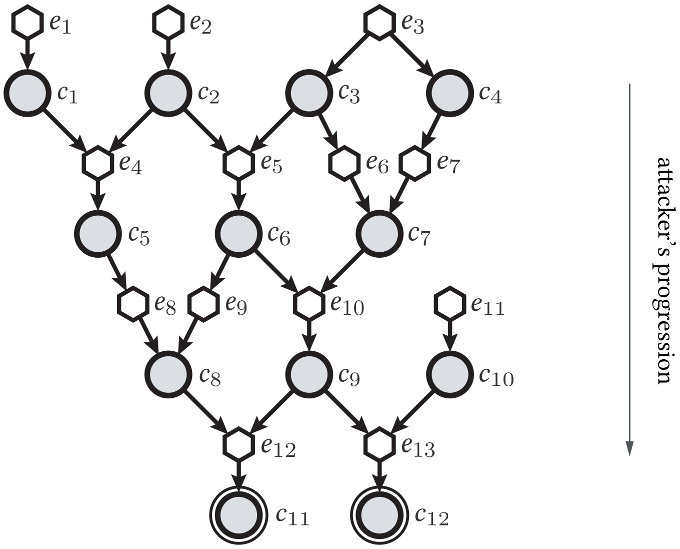


Figure 6.2: A sample condition dependency graph. The above dependency graph $\mathcal{H} = (\mathcal{N}, \mathcal{E})$ consists of $n_c = 12$ security conditions and $n_e = 13$ exploits (in the form of hyperedges). Initial exploits, $\mathcal{E}_0 = \{e_1, e_2, e_3, e_{11}\}$, where $e_1 = (\varnothing, \{c_1\})$, $e_2 = (\varnothing, \{c_2\})$, $e_3 = (\varnothing, \{c_3, c_4\})$, $e_{11} = (\varnothing, \{c_{10}\})$ and exploits $e_4 = (\{c_1, c_2\}, \{c_5\})$, $e_5 = (\{c_2, c_3\}, \{c_6\})$, $e_6 = (\{c_3\}, \{c_7\})$, $e_7 = (\{c_4\}, \{c_7\})$, $e_8 = (\{c_5\}, \{c_8\})$, $e_9 = (\{c_6\}, \{c_8\})$, $e_{10} = (\{c_6, c_7\}, \{c_9\})$, $e_{12} = (\{c_8, c_9\}, \{c_{11}\})$, $e_{13} = (\{c_9, c_{10}\}, \{c_{12}\})$. We represent the graph in a layered structure in which preconditions are drawn above postconditions, *e.g.* exploit e_4 has preconditions $\{c_1, c_2\}$ and a single postcondition $\{c_5\}$. Goal conditions, $\mathcal{N}^g = \{c_{11}, c_{12}\}$, are represented by double-encircled nodes.

We assume that the dependency graph has already been constructed for the system (using vulnerability analysis tools such as the TVA tool of [Jajodia et al., 2005]) and instead focus on the formulation and solution of a real-time (dynamic) defense problem, using the dependency graph to describe the progression of the attacker.

6.2.2. The Notion of a Security State

A primary objective of our dynamic security model is to quantify the level of security of the system over time. To this end, we define a *security state* to represent the current level of progression of the attacker in the system. The current security state, denoted by $s_t \subseteq \mathcal{N}$, is defined to be the set of currently enabled security conditions. Since an enabled condition is interpreted as the attacker having a particular capability, the current security state, s_t, describes the set of capabilities of the attacker.

The monotonicity assumption on the attacker's behavior implies a notion of *feasibility* for the security states, defined formally below.

Definition 6.2.1 (Feasible Security State). *A security state, $s \subseteq \mathcal{N}$, is called a* feasible security state *if for every condition $c_j \in s$, there exists at least one exploit $e_i = (\mathcal{N}_i^-, \mathcal{N}_i^+) \in \mathcal{E}$ such that $c_j \in \mathcal{N}_i^+$ and $\mathcal{N}_i^-, \mathcal{N}_i^+ \subseteq s$.*

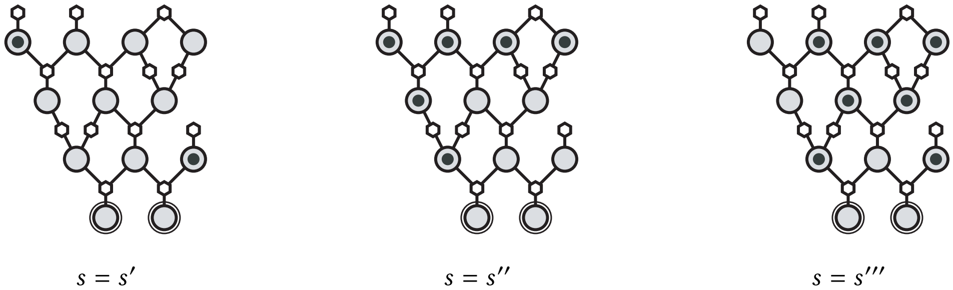
That is, in order for a security state to be feasible, every enabled condition must have been enabled through an exploit and all preconditions and postconditions of the associated exploit must also be enabled. An implicit assumption behind the feasibility condition is that our model for exploits is complete, in the sense that our model is not missing any exploits that would allow the attacker to enable security conditions.[‖] Fig. 6.3 illustrates a few feasible security states.

The state space of the dynamic security model consists of all feasible security states, denoted by $\mathcal{S} = \{s_1, \ldots, s_{n_s}\}$. We do not have a closed-form expression for the number of feasible states n_s; however, as discussed in Section 6.3, the proposed online defense algorithm does not require one to construct the entire state space.

6.2.3. Evolution of the Security State

The security state evolves probabilistically as a function of both the defender's and attacker's actions. In a given iteration of our problem, the defender is assumed to act first, taking actions that interfere with the attacker's progression through the system by dynam-

[‖] Relaxing this assumption amounts to including nodes in s that are not associated with any hyperedge in $\mathcal{E}$, meaning that they can become enabled via an unknown influence. The inclusion of these *leaky nodes* greatly increases the state space and is not considered in this work.



$s = s'$ $s = s''$ $s = s'''$

Figure 6.3: A collection of feasible security states for the graph $\mathcal{H} = (\mathcal{N}, \mathcal{E})$ of Fig. 6.2. Enabled security conditions are represented by shaded nodes. Notice that for each feasible security state, there is a path of exploits in $\mathcal{E}$ from enabled root condition(s) to each enabled (non-root) condition such that all preconditions and postconditions of the respective exploits are enabled.

ically modifying the *attack surface* (the collection of various pathways that the attacker can use to infiltrate the system). The attacker then uses its set of current capabilities to attempt exploits, the dynamics of which are dictated by its (private) attack strategy. Finally, the attempted exploits that end up succeeding determine the transition to the next security state.

Defender's Actions

The defender is assumed to select actions that have the effect of restricting the normal network configuration (such as the network connectivity or active services). Performing such system modifications has the effect of *blocking* the exploits that depend on the network elements that were modified. As a simple example, some exploits depend on the existence of a connection between hosts via a specific port. By blocking this port between the hosts, we are able to block the corresponding exploits that depend on the port being open, preventing the attacker from using these exploits to progress through the system.

In reality, the defender is not able to block individual exploits at will. The system modifications involved in blocking one exploit will, in general, block multiple exploits in the system (*e.g.* blocking a port or disabling a service). On the other hand, some exploits may not be able to be blocked by any of the defender's available system changes (*e.g.* an exploit of a local software vulnerability that results in the escalation of attacker's privilege on a specific host). This coarseness in the ability to block exploits translates into the defender having limited control over the attacker's progression through the system, a characteristic which is captured in our definition of the defender's set of actions (described below).

Formally, the defender is assumed to have access to $n_u + 1$ defense actions, represented by the set $\mathcal{U} = \{u^0, u^1, \ldots, u^{n_u}\}$. The defense action u^0 represents the null action and corresponds to the defender not blocking any exploits, allowing the system (and attacker) to operate uninterrupted. Each of the n_u remaining defense actions, u^i, $i = 1, \ldots, n_u$, corresponds to a set of system modifications that restrict the normal network configuration, such as restricting the network connectivity (*e.g.* by blocking a port between some hosts) or the set of active services, and can be associated with blocking a specific set of exploits, denoted by $\mathcal{B}(u^i) \subseteq \mathcal{E}$. Notice that the defender does not, in general, have the ability to block individual exploits, instead it must select a defense action $u \in \mathcal{U}$ which in turn induces a set of blocked exploits $\mathcal{B}(u) \subseteq \mathcal{E}$.

Each defense action $u \in \mathcal{U}$ induces system modifications that interfere with the progression of the attacker but also, unavoidably, limit the availability of the system to trusted users. It is the goal of the defense scheme to optimally balance this tradeoff. As described in Section 6.2.5, a cost is assigned to each defense action u in order to capture its impact to availability. Combined with the assignment of costs for undesirable security states, the defender is able to specify actions that limit the attacker while minimizing the negative impact to availability.

Threat Model

It is assumed that there is a single attacker attempting to infiltrate the system. At any given time-step, the attacker attempts to enable security conditions by performing exploits, in hopes of increasing its set of capabilities and allowing it to progress through the system. The specific nature of the attacker's progression is given by its (private) strategy, dictated by one of a finite set of attacker *types*. As will be described in the remainder of Section 7.2, the attacker type dictates the dynamics of both the security state and observation processes. Lastly, the attacker is assumed to be monotone. As discussed earlier, the monotonicity assumption states that the success of a previous exploit will not interfere with the success of a future exploit. In the context of the proposed model, this implies that once the attacker enables a security condition, it remains enabled.

Formally, for a given security state s_t, the set of exploits that the attacker can attempt, termed the *available exploits*, is described by the set $\mathcal{E}(s_t)$. This set represents the complete set of exploits that are available from state s_t. The attacker does not necessarily know all of the elements in this set; $\mathcal{E}(s_t)$ simply represents what can be attempted using the capabilities described by s_t. The set of available exploits is given by

$$\mathcal{E}(s_t = s) = \left\{ e_i = (\mathcal{N}_i^-, \mathcal{N}_i^+) \in \mathcal{E} \mid \mathcal{N}_i^- \subseteq s, \mathcal{N}_i^+ \nsubseteq s \right\}. \tag{6.1}$$

In order for an exploit $e_i = (\mathcal{N}_i^-, \mathcal{N}_i^+)$ to be available to the attacker, it must satisfy two requirements. The first requirement, $\mathcal{N}_i^- \subseteq s$, states that all of the exploit's preconditions must be satisfied in the current security state. The second requirement, $\mathcal{N}_i^+ \nsubseteq s$, states that the exploit's postconditions must not all be satisfied. This latter requirement arises from the assumption that the attacker will not perform redundant exploits. This assumption is reasonable since the attacker will not gain any new capabilities by performing such exploits and will only increase its chances of being detected (discussed further in Section 6.2.4). The caption of Fig. 6.4 describes the set of available exploits, for a given security state s_t, in the example condition dependency graph $\mathcal{H}$ of Fig. 6.2.

The specific strategy that the attacker employs is dictated by its type. The attacker is assumed to be one of n_a types, represented by the set $\Phi = (\varphi_1, \ldots, \varphi_{n_a})$. Each type $\varphi_i \in \Phi$ corresponds to a set of *conditional attack probabilities* over the exploits, $\alpha(\varphi_i, s_t, u_t) = \big(\alpha_{e_1}(\varphi_i, s_t, u_t), \ldots, \alpha_{e_{n_e}}(\varphi_i, s_t, u_t)\big)$, specifying the likelihood that the attacker will attempt each of the available exploits from the current security state s_t under defense action u_t. The conditional attack probability for a given exploit e_j is given by

$$\alpha_{e_j}(\varphi_i, s_t, u_t) = \begin{cases} \overline{\alpha}_{e_j}(\varphi_i) & \text{if } e_j \in \mathcal{E}(s_t) \setminus \mathcal{B}(u_t) \\ \underline{\alpha}_{e_j}(\varphi_i) & \text{if } e_j \in \mathcal{E}(s_t) \cap \mathcal{B}(u_t) \\ 0 & \text{if } e_j \notin \mathcal{E}(s_t) \end{cases} . \tag{6.2}$$

By partitioning the set of available exploits into two components, the threat model describes how an attacker may modify its strategy based on the defender's action. Specifically, available exploits that are not blocked by the current defense action, $\mathcal{E}(s_t) \setminus \mathcal{B}(u_t)$, are attempted with probability $\overline{\alpha}_{e_j}(\varphi_i)$, whereas exploits that are blocked by the current defense action, $\mathcal{E}(s_t) \cap \mathcal{B}(u_t)$, are attempted with probability $\underline{\alpha}_{e_j}(\varphi_i)$. Exploits that are not available in the current security state are not attempted.

Constructing the threat model in such a way allows one to encode various levels of attacker knowledge. For example, if an attacker of type φ_i is not able to recognize that exploit e_j is blocked under a give defense action $u_t = u$, then $\overline{\alpha}_{e_j}(\varphi_i) = \underline{\alpha}_{e_j}(\varphi_i)$, reflecting the fact that the attacker is unable to modify its attack probability based on the defender's action. On the other hand, if the attacker knows with certainty that exploit e_j has been blocked by the defender, then setting $\underline{\alpha}_{e_j}(\varphi_i) = 0$ reflects that the attacker would not attempt it. The threat model can also capture intermediate cases where the attacker has partial information and may attempt exploits that it believes are not blocked with a higher probability, *i.e.*

$\overline{\alpha}_{e_j}(\varphi_i) \geq \underline{\alpha}_{e_j}(\varphi_i)$.

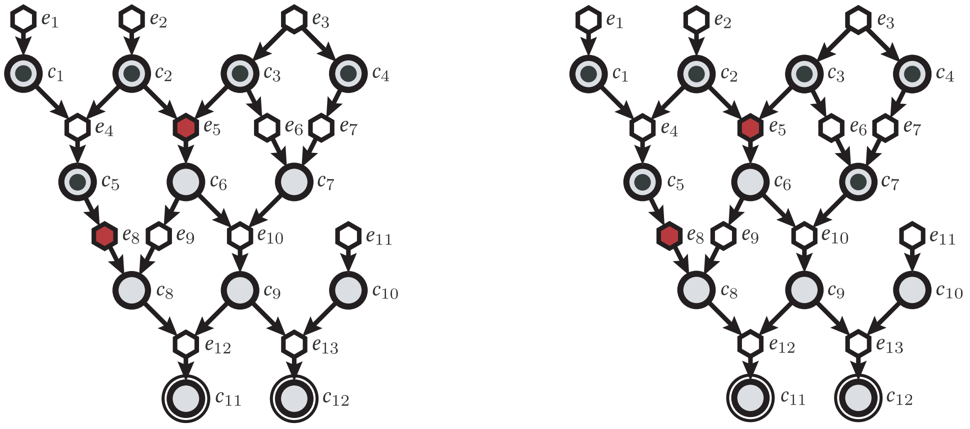
Security State Dynamics

For any given iteration, the defender first chooses a defense action, $u_t = u \in \mathcal{U}$, in turn blocking a set of exploits, $\mathcal{B}(u) \subseteq \mathcal{E}$. Next, given the current security state $s_t \in \mathcal{S}$ and defense action $u_t = u$, the attacker attempts a collection of available exploits according to its own private strategy, $\alpha(\varphi_i, s_t, u_t)$. Each of the attempted exploits succeeds with a *conditional probability of success*. The probability of success models the fact that attacks do not succeed with certainty (potentially due to the inherent difficulty in carrying out the attack or the existence of defenses already in place). The probabilities are assumed to depend upon the attacker's type; this dependency arises from the fact that some attackers may possess greater knowledge of the exploit or be able to expend more resources. The set of conditional success probabilities is given by $\beta(\varphi_i, u_t) = \big(\beta_{e_1}(\varphi_i, u_t), \ldots, \beta_{e_{n_e}}(\varphi_i, u_t)\big)$, where the probability of success for a given exploit e_j is given by

$$\beta_{e_j}(\varphi_i, u_t) = \begin{cases} \beta_{e_j}(\varphi_i) & \text{if } e_j \notin \mathcal{B}(u_t) \\ 0 & \text{if } e_j \in \mathcal{B}(u_t) \end{cases}. \tag{6.3}$$

Exploits that are blocked by the defender do not succeed. The exploit attempts that are successful enable the corresponding set of postconditions, forming the updated security state $s_{t+1} \in \mathcal{S}$. Fig. 6.4(b) illustrates a possible successor state s_{t+1} for a given state s_t, type φ_t, and defense action u_t.

In summary, the security state dynamics can be described by a controlled Markov chain, where the control is the defense action. The transition matrix of the Markov chain, for a given defense action u and attacker type φ_l, is $P^u(\varphi_l)$ with elements $p^u_{ijl} = P(S_{t+1} = s_j \mid S_t = s_i, \Phi_t = \varphi_l, U_t = u)$ (the analytical expression for this probability is given by Eq. (C.1) in the



(a) Current security state

(b) Possible successor state

Figure 6.4: Sample evolution of the security state for a given state-type-action triple (s_t, φ_t, u_t): **(a)** Consider the security state $s_t = \{c_1, c_2, c_3, c_4, c_5\}$ and defense action $u_t = u$ such that $\mathcal{B}(u) = \{e_5, e_8\}$ (blocked exploits are shown by shaded hyperedges). By Eq. (6.1), the set of available exploits is $\mathcal{E}(s_t) = \{e_5, e_6, e_7, e_8\}$; **(b)** The attacker attempts each exploit in $e_i \in \mathcal{E}(s_t)$ with a probability of attack $\alpha_{e_i}(\varphi_t, s_t, u_t)$ as described in Eq. (6.2). Each attempted exploit e_i that does not lie within the set of blocked exploits, $\mathcal{B}(u_t) \subseteq \mathcal{E}$, succeeds with a probability of success $\beta_{e_j}(\varphi_i)$ as described in Eq. (6.3). In this example, only exploit e_6 succeeded and thus $s_{t+1} = s_t \cup \mathcal{N}_6^+ = s_t \cup \{c_7\}$.

appendix).[**] Note that defense actions only influence the attacker's progression. Blocking an exploit that already has all of its postconditions enabled does not disable any of the exploit's postconditions. An analogy to the physical security domain is useful: Consider an intruder attempting to break into a building to access a safe. If the intruder has already successfully broken through the front door, then barricading the door will have no effect on the attacker's ability to access the safe. However, securing the door before the attacker reaches it will prevent the attacker from using that entry point, forcing it to use another path, in turn increasing the attacker's effort and decreasing the likelihood of the safe being compromised.

[**] The proposed security model also allows for the underlying type to vary in time according to a Markov chain (with transition matrix Q); however, for simplicity we consider a fixed (albeit unknown) underlying attacker type.

6.2.4. The Defender's Information

The defender lacks certainty of both the current security state and the underlying strategy of the attacker and must infer/learn both using a stream of noisy security information. The security information comes in the form of a sequence of *security alerts* generated by an intrusion detection system (IDS) as the attacker attempts exploits and progresses through the system (see Fig. 6.1). These security alerts are noisy, suffering from both missed detections (the IDS not seeing an exploit attempt) and false alarms (the IDS generating alerts when no attempt has occurred, *e.g.*, alerts generated by legitimate network traffic).

Let $\mathcal{Z} = \{z_1, z_2, \ldots, z_{n_z}\}$ represent the finite set of security alerts that may be generated by the IDS. Each exploit $e_i \in \mathcal{E}$, if attempted, has an associated set of alerts that can be generated, given by the set $\mathcal{Z}(e_i) = \{z_{\mathcal{A}_i(1)}, z_{\mathcal{A}_i(2)}, \ldots, z_{\mathcal{A}_i(a_i)}\} \in \mathscr{P}(\mathcal{Z})$, where $\mathcal{A}_i$ is the set of a_i alert indices from the set $\mathcal{A} = \{1, 2, \ldots, n_z\}$ and $\mathscr{P}(\mathcal{Z})$ is the power set of $\mathcal{Z}$. In general, more than one exploit can generate the same alert, that is $\mathcal{Z}(e_i) \cap \mathcal{Z}(e_j) \neq \varnothing$ for $e_i \neq e_j$. Also, some exploits may not generate any alerts, that is, $\mathcal{Z}(e_i) = \varnothing$ for some $e_i \in \mathcal{E}$ (such exploits are termed *stealthy*).

The IDS generates the security alerts probabilistically, based on detected exploit activity and false alarms, the statistics of which depend upon the underlying strategy of the attacker. Advanced attackers may be able to craft their attacks such that they are less likely to trigger security alerts or, alternatively, influence the false alarm rate of specific alerts to mask their true progression through the system. To capture this dependency, the security model allows for the *probabilities of detection* (the likelihood of seeing an alert given an exploit attempt) and the *probabilities of false alarm* (the likelihood of seeing an alert in the absence of an exploit attempt) to depend on the attacker's type. For an attacker of type φ_l, an attempt of exploit e_i will generate the alerts $\mathcal{Z}(e_i) = \{z_{\mathcal{A}_i(1)}, z_{\mathcal{A}_i(2)}, \ldots, z_{\mathcal{A}_i(a_i)}\}$ with corresponding probabilities of detection $\delta_{ij}(\varphi_l)$, $j \in \mathcal{A}_i$. Similarly, the probability of false alarm for each alert $z_i \in \mathcal{Z}$, under type φ_l, is dictated by $\zeta_i(\varphi_l)$. The vector of security alerts received by the defender at time $t+1$, denoted by $y_{t+1} \in \mathcal{Y} = \{0,1\}^{n_z}$, consists of all security alerts triggered during the given iteration.

Using the received security alerts the defender constructs a belief, denoted by π_t, that summarizes its uncertainty over both the security state and the attacker type. This belief (or *information state* [Åström, 1965, Kumar & Varaiya, 1986]) is constructed using all of the defender's available information at time t: the (distribution over the) initial security state and attacker type, the history of all defense actions from time 0 to time $t-1$, and all observations (security alerts) from time 0 to t, denoted by $h_t = (\pi_0, u_0, y_0, \ldots, u_{t-1}, y_t)$. The belief represents the joint probability distribution over security states and attacker types, and takes the form of a matrix, defined as

$$\pi_t = \begin{bmatrix} \pi_t^{1,1} & \pi_t^{1,2} & \cdots & \pi_t^{1,n_a} \\ \pi_t^{2,1} & \pi_t^{2,2} & \cdots & \pi_t^{2,n_a} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_t^{n_s,1} & \pi_t^{n_s,2} & \cdots & \pi_t^{n_s,n_a} \end{bmatrix} \in \Delta(\mathcal{S} \times \Phi)$$

where $\pi_t^{il} = P(S_t = s_i, \Phi_t = \varphi_l \mid H_t = h_t)$ is the likelihood that s_i is the true security state and φ_l is the true type given the realized information h_t. The space $\Delta(\mathcal{S} \times \Phi)$ is the probability simplex over the state-type space $\mathcal{S} \times \Phi$. Notice that π_t is a doubly-stochastic matrix for each t; each row represents a probability mass function over the type space for a given state and each column represents a probability mass function over the space of security states for a given type.

The defender maintains the belief matrix over time, updating it as new information, consisting of the current defense action u_t and new observation y_{t+1}, is revealed. For a given defense action $u_t = u$ and observation $y_{t+1} = y_k$, the belief matrix update is defined as $\pi_{t+1} = [\tau_{jm}\big(\pi_t, u, y_k\big)]_{\varphi_m \in \Phi, s_j \in \mathcal{S}}$ where the (j,m)'th update function, $\tau_{jm}\big(\pi_t, u, y_k\big) =$

$P(S_{t+1} = s_j, \Phi_{t+1} = \varphi_m \mid U_t = u, Y_{t+1} = y_k, \Pi_t = \pi_t)$, is given by

$$\pi_{t+1}^{jm} = \tau_{jm}\Big(\pi_t, u, y_k\Big) = \frac{p_{jm}^u(\pi_t) r_{jk}^u(\pi_t)}{\sigma(\pi_t, u, y_k)}. \tag{6.4}$$

The above terms are defined as follows

$$\begin{aligned}
p_{jm}^u(\pi_t) &= P(S_{t+1} = s_j, \Phi_{t+1} = \varphi_m \mid U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} \pi_t^{il} p_{ijl}^u q_{lm} && (6.5) \\
r_{jk}^u(\pi_t) &= P(Y_{t+1} = y_k \mid S_{t+1} = s_j, U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} \pi_t^{il} r_{ijkl}^u && (6.6) \\
\sigma(\pi_t, u, y_k) &= P(Y_{t+1} = y_k \mid U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_j \in \mathcal{S}, \varphi_m \in \Phi} r_{jk}^u(\pi_t) p_{jm}^u(\pi_t) && (6.7)
\end{aligned}$$

where p_{ijl}^u is the probability of transitioning from state s_i to s_j under defense action u and attacker type φ_l, q_{lm} is the probability of transitioning between types (note that we assume that $q_{lm} = 1$ if $l = m$, zero otherwise, as mentioned in Section 6.2.3), and $r_{ijkl}^u = P(Y_{t+1} = y_k \mid S_{t+1} = s_j, S_t = s_i, \Phi_t = \varphi_l, U_t = u)$ is the probability that the IDS generates observation y_k given a transition from state s_i to s_j under defense action u and attacker type φ_l. Additional details regarding the derivation of the belief update equations, as well as analytical expressions for p_{ijl}^u and r_{ijkl}^u, can be found in the appendix.

The belief at any given time represents the *defender's view of the attacker's current capabilities and true strategy.* The trajectory of beliefs, $(\pi_0, \pi_1, \pi_2, \ldots)$, describes how this view changes over time. As evidenced by Eq. (6.4), the trajectory of beliefs (given an initial belief) is defined by the sequence of defense actions and observations (security alerts). Since security alerts are triggered probabilistically by exploit attempts and background events, the presence of an alert does not necessarily mean that the attacker is progressing through the system. That is, an exploit attempt may have triggered an alert but may not have succeeded, or an alert may have been triggered via a false alarm. Similarly, the absence of an alert may mean that an exploit was in fact attempted (and successful), but didn't trigger an alert (due to a missed detection or a stealthy exploit). Since the current belief π_t assigns mass to security states (and attacker types) that are consistent with the available information, the belief trajectory may assign mass to worsening security states even in the cases where the underlying security state is unchanging or no alerts are generated. This characteristic highlights the importance of information in our model, reflecting that the defender's imperfect observations of the security state and attacker type contribute to a more pessimistic view of the system's security over time.

6.2.5. Assignment of Costs

In many systems, the cyber network needs to remain (at least partially) operational while subject to an attack. The defender thus has two objectives: i) maintaining the availability of the system, and ii) keeping the attacker away from goal conditions. These two factors are largely in opposition of each other. If the defender were only concerned with maintaining the availability of the system, it would not perform any system modifications, leaving the system to run uninterrupted and in turn not interfering with the progression of the attacker. On the other hand, if the defender were just concerned with preventing the attacker from reaching goal conditions, it would immediately execute aggressive system changes in order to block as many exploits as possible and maximally disrupt the attacker's progression through the system. Unfortunately, this latter option is clearly very costly to the availability of the system. It is evident that one must strike a trade-off between these two extremes of availability and security.

In order to quantify this trade-off, we construct a cost function that takes into account both the quality of the current security state and the negative impact to availability of each defense action. Specifically, we assign a *security cost*, $c_s : \mathcal{S} \times \Phi \rightarrow \mathcal{R}$, to capture the cost of the system being in various security states $s \in \mathcal{S}$ under different attacker types as well as an *availability cost*, $c_u : \mathcal{U} \rightarrow \mathcal{R}$, for each defense action that is deployed. Using the definition of a goal condition at the end of Section 6.2.1, we can define the notion of a goal state.

Definition 6.2.2 (Goal state). *A* goal state *is defined as a security state $s \in \mathcal{S}$ that contains one or more goal conditions, that is, there exists some $j \in s$ such that $j \in \mathcal{N}^g$.*

We denote the space of all goal states by $\mathcal{S}^g \subseteq \mathcal{S}$. Goal states are undesirable from the perspective of the defender and are thus assigned a higher cost than non-goal states, that is, $0 \leq c_s(s', \varphi) \leq c_s(s'', \varphi) < \infty$ for $s' \notin \mathcal{S}^g$, $s'' \in \mathcal{S}^g$, $\varphi \in \Phi$. Although not a requirement, we can impose the additional property that for any two security states $s', s'' \in \mathcal{S}$ where $s' \subseteq s''$, we have $c(s', \varphi) \leq c(s'', \varphi)$, reflecting the fact that if the attacker has enabled more conditions, it should be more costly for the defender. To model the availability factor, we assign an availability cost for each defense action, denoted by $c_u(u')$. Recall that each defense action $u' \in \mathcal{U}$ is a collection of system modifications. Some combinations of system modifications may have little to no impact to availability while other combinations may render important elements of the underlying system unavailable. The assignment of the costs $c_u(u')$, for each $u' \in \mathcal{U}$, allows one to incorporate such information (combinations of system modifications that severely impact availability should be assigned a very high cost). We assume that $0 \leq c_u(u') < \infty$ for every $u' \in \mathcal{U}$. The cost for taking defense action u_t in security state s_t under attacker type φ_t is defined as

$$c(s_t, \varphi_t, u_t) = wc_s(s_t, \varphi_t) + (1 - w)c_u(u_t) \tag{6.8}$$

where $0 \leq w \leq 1$ is a weighting term that allows the defender to specify which factor is more important, where $w = 0$ ($w = 1$) corresponds to only being concerned with availability (resp. security).

6.2.6. Defender's Problem

The defender wishes to determine an optimal defense action to deploy for any belief that it may encounter. The decision rule determining this action is termed a *defense policy* and is represented by the function $\gamma : \Delta(\mathcal{S} \times \Phi) \to \mathcal{U}$, mapping a belief matrix $\pi \in \Delta(\mathcal{S} \times \Phi)$ to a defense action $u \in \mathcal{U}$. The problem of determining γ can be cast as a POMDP, represented by problem (P) below.

$$\min_{\gamma \in \Gamma} \mathbb{E}^{\gamma} \left\{ \sum_{t=0}^{\infty} \rho^t c(\Pi_t, U_t) \mid \Pi_0 = \pi_0 \right\} \tag{P}$$

$$\text{subject to } U_t = \gamma(\Pi_t) \tag{P-1}$$

$$\Pi_{t+1} = \tau(\Pi_t, U_t, Y_{t+1}) \tag{P-2}$$

where Γ is the space of admissible defense policies and $0 < \rho < 1$ is the discount factor. The function $c(\pi_t, \varphi_t, u_t)$ represents the expected cost for being in belief state $\Pi_t = \pi_t$ when defense action $U_t = u_t$ is selected and is defined as $c(\pi_t, u_t) = \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} \pi_t^{il} c(s_i, \varphi_l, u_t)$ where $c(s, \varphi, u)$ is the state-action cost function defined in Eq. (6.8). The current action U_t must be generated according to the defense policy γ, as demonstrated by constraint (P-1), and the next belief Π_{t+1} must obey the update $\tau(\Pi_t, U_t, Y_{t+1})$, constraint (P-2).

The solution to problem (P) is an optimal defense policy, denoted by $\gamma^* \in \Gamma$, which specifies an optimal defense action for every possible belief $\pi \in \Delta(\mathcal{S} \times \Phi)$ that the defender can possess. Following the optimal policy results in the minimum expected discounted cost over the infinite time-horizon, $t = 0, 1, \ldots$. In other words, taking into account all uncertainty in the problem, the defense policy γ^* generates actions that achieve the desired tradeoff as dictated by the cost function in Section 6.2.5.

6.3. Computation of Defense Policies

While the embedding of a state space on the dependency graph allows for one to accurately quantify the level of progression of the attacker, the high dimensionality of the resulting defense problem leads to significant scalability concerns. One approach to solving the defense problem is to adopt an *offline* POMDP solver. Such solvers aim to explicitly solve the problem by computing the optimal action for every belief that can be encountered, prior to runtime. In spite of the fact that significant improvements have been made in the efficiency of offline solvers in recent years, *e.g.* [Kurniawati et al., 2008], the requirement to specify an action for every possible encountered scenario often leads to an intractable problem. *Online* solvers represent an alternate paradigm in which one only considers the possible future scenarios from the current belief, constructing a local policy during runtime. Online methods interleave the computation and execution (runtime) phases of a policy [Ross et al., 2008], yielding a much more scalable approach than offline methods, making them a more natural fit for obtaining a solution to the defense problem.

The proposed algorithm for computing defense policies, which we term the *online defense algorithm*, is based on an existing online solver developed by Silver & Veness [Silver & Veness, 2010], termed the Partially Observable Monte-Carlo Planning (POMCP) algorithm. While no existing algorithm is immediately applicable for computing defense policies, the POMCP algorithm requires the fewest modifications to achieve efficient computation.

6.3.1. The Online Defense Algorithm

The online defense algorithm is a heuristic search algorithm for determining defense actions in real-time as the attacker progresses through the system and security alerts are generated. The algorithm consists of two main stages: an *action selection* step and a *belief update* step. The action selection step of the online defense algorithm is similar to that of the standard POMCP algorithm (details pertaining to the specific operation of POMCP, as well as pseudocode for the algorithm, can be found in [Silver & Veness, 2010]). The belief update step has been modified by taking advantage of the structure of the observation process, enabling computation in large-scale domains.

The action selection stage of the online defense algorithm operates by performing Monte-Carlo simulations from the current belief in order to estimate the quality of various defense actions. Each simulation consists of a call to a *generative model*, shown in Fig. 6.5. Specifically, a simulation begins by sampling a state-type pair, (s, φ), from the current belief matrix (approximated by a finite collection of state-type pairs, described in more detail in the following paragraph), and coupled with a given defense action, generates a successor state and type, as well as an observation and cost, $(s', \varphi', y, c) \sim \mathcal{G}(s, \varphi, u)$. Through successive sampling from the current belief and calls to the generative model, a search tree of histories is constructed, as shown in Fig. 6.6. Due to the partial observability of the underlying process,

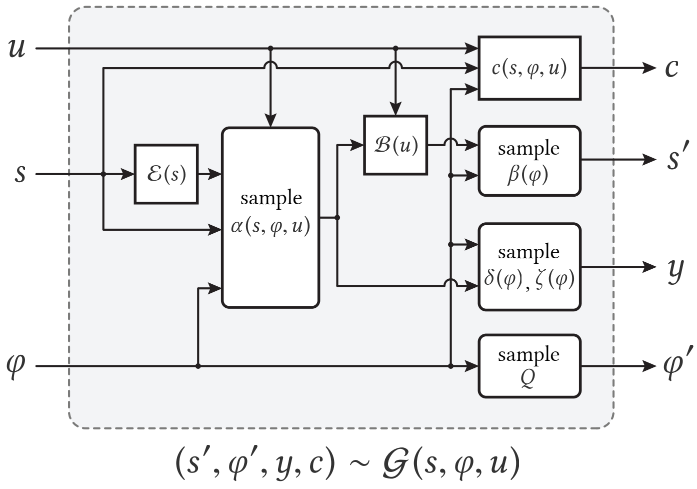

u
s
φ
$\mathcal{E}(s)$
sample $\alpha(s, \varphi, u)$
$\mathcal{B}(u)$
$c(s, \varphi, u)$
sample $\beta(\varphi)$
sample $\delta(\varphi), \zeta(\varphi)$
sample Q
c
s'
y
φ'
$(s', \varphi', y, c) \sim \mathcal{G}(s, \varphi, u)$

Figure 6.5: The generative model for the dynamic security model. For a given state-type-action triple (s, φ, u), the generative model first determines the set of available exploits, $\mathcal{E}(s)$, then, taking into account the effect of the defender's action and attacker type, samples the probability distributions of the problem (probabilities of attack and success for exploits, and probabilities of detection and false alarm for observations) in order to generate a next state $s_{t+1} = s'$, updated type $\varphi_{t+1} = \varphi'$, an observation $y_{t+1} = y$, and a cost c.

the search tree consists of nodes representing histories, where branches of the tree originating from the current history represent future possible histories. Each branch begins with the selection of a defense action, which is selectively sampled using a multi-armed bandit rule, termed UCB1 [Auer et al., 2002], in order to optimally balance exploitation and exploration. That is, selecting presumably promising actions in order to decrease their estimation error must be balanced with checking other actions in order to rule out better alternatives. The error associated with each defense action's quality estimate decreases as the number of simulations (and the size of the search tree) grows. The online defense algorithm continues to perform simulations for the given history node, progressively expanding the tree, until a maximum number of simulations, n_{sim}, has been reached. The defense action that has the lowest estimated cost is then taken, termed the real-world action, denoted by u_r, and a real-world observation, denoted by y_r, is recorded. The relevant branches of the search tree are identified (shown by the blue/shaded path to node h' in Fig. 6.6), the remaining tree is pruned, and a new root node is specified as the current history.

Once an updated history h' is realized, the defender's belief must be updated. Due to the computational complexity associated with updating the belief matrix analytically (see Eq. (6.4) and the appendix), the defender maintains a belief approximation, denoted by B_t, consisting of n_k state-type pairs, termed particles. The update of the belief approximation under the standard POMCP algorithm involves making multiple calls to the generative model in order to obtain samples (s', y) where y exactly matches the real-world observation y_r, at which point s' is accepted into the updated belief set B_{t+1} (repeating until n_k particles have been added). Instances of the security model with large observation spaces allow for scenarios where the sampled observation rarely matches the real-world observation, preventing the belief from being updated. To address this issue, we propose a modified belief update that takes advantage of the structure of the observations, that is, how security alerts are generated as a function of the security state and type. Instead of checking if the sample

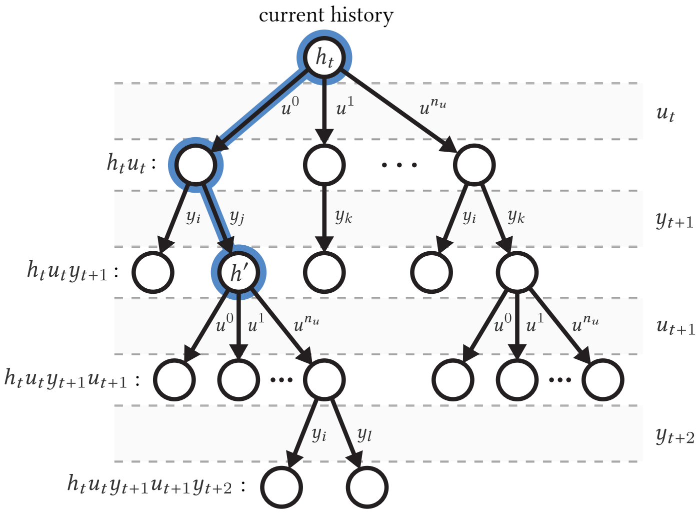


Figure 6.6: A search tree of histories. Each node in the search tree represents a history. The root node represents the current history from which simulations begin. Each descendant of the root node represents a possible future history, for example, a possible realized history for $h_t u_t y_{t+1} u_{t+1}$ is $h_t u^0 y_j u^1$. After a real-world action is taken, *e.g.* $u_r = u^0$, and a real-world observation is received, *e.g.* $y_r = y_j$, the history is updated to h' (represented by the blue/shaded path).

observation matches the real-world observation for every alert $z_i \in \mathcal{Z}$, the proposed belief update only checks if the alerts agree over a security state dependent subset of elements $z_i \in \mathcal{Z}(s) = \cup_{e \in \mathcal{E}(s)} \mathcal{Z}(e)$ and probabilistically accepts the particle if this modified condition is satisfied. The set $\mathcal{Z}(s)$ represents the set of alerts that can be generated by exploit attempts; alerts not in $\mathcal{Z}(s)$, *i.e.* any alert in $\bar{\mathcal{Z}}(s) = \mathcal{Z} \setminus \mathcal{Z}(s)$, cannot be generated by the attempt of any exploit available in state s, as dictated by Eq. (6.1). The rationale for restricting the comparison to the elements $\mathcal{Z}(s)$ is due to the fact that these are the only alerts that are informative for a change in the underlying state. The remaining alerts $\bar{\mathcal{Z}}(s)$ must have been triggered by false alarms under the current state s. Observations that pass the modified test are accepted into the updated belief with a probability of acceptance that depends on the security state and attacker type, $p_a(s, \varphi)$. The probability of acceptance is dictated by the likelihood that the state-type pair (s, φ) could have generated the real-world observation and is defined as $p_a(s,\varphi) = \bar{p}_a(s,\varphi)/d$ where

$$\bar{p}_a(s,\varphi) = \Bigg(\prod_{i\in\mathcal{I}(y_r^{\bar{\mathcal{Z}}(s)}=1)} \zeta_i(\varphi)\Bigg)\Bigg(\prod_{i\in\mathcal{I}(y_r^{\bar{\mathcal{Z}}(s)}=0)} (1-\zeta_i(\varphi))\Bigg)$$

and $d = \max_{(s,\varphi)\in B_t} \bar{p}_a(s,\varphi)$ is a normalization term. The set $\mathcal{I}(y_r^{\bar{\mathcal{Z}}(s)} = 1)$ represents the indices of the alerts $z_i \in \bar{\mathcal{Z}}(s)$ where $y_r^i = 1$ is true (analogously for $\mathcal{I}(y_r^{\bar{\mathcal{Z}}(s)} = 0)$). The normalized probability of acceptance, $p_a(s,\varphi)$, ensures that particles are accepted into the updated belief more frequently than the standard POMCP belief update while ensuring that the relative mass under the modified belief procedure agrees with what would be achieved under the standard belief update. The pseudo code for the modified belief update is given in Algorithm 3 below.

Algorithm 3 – Modified Belief Update

```
Initialize: $n_k$, $B_{t+1} = \varnothing$, numAdded = 0;
procedure ModifiedBeliefUpdate($B_t$,$u_r$,$y_r$)
    while numAdded < $n_k$ do
        $(s,\varphi) \sim B_t$
        $(s',\varphi',y,-) \sim \mathcal{G}(s,\varphi,u_r)$
        if $y^{\mathcal{Z}(s)} = y_r^{\mathcal{Z}(s)}$ then                ▷ If alerts $\mathcal{Z}(s)$ match
            $B_{t+1} \leftarrow B_{t+1} \cup \{s',\varphi'\}$ with probability $p_a(s,\varphi)$
            numAdded ← numAdded + 1
        end if
    end while
end procedure
```

In addition to the modified belief update procedure, a heuristic cost assignment can further improve the scalability of the online defense algorithm. A key bottleneck for tree-based heuristic search algorithms in large-scale domains is the rate at which the search tree grows as a function of the depth from the root node, termed the *branching factor*. Problem instances with many actions and observations result in search trees with large branching factors, preventing the search algorithm from being able to search beyond a small depth, resulting in a poor quality, myopic policy. To avoid this, we can assign non-zero costs to security states that are *close* to goal states. A simple procedure for such a cost assignment is to assign higher costs to states that require fewer successful exploits to reach a goal state. Such a heuristic cost assignment makes simulations more informative, decreasing the required search depth (and simulations) and resulting in more effective defense policies.

Using the above ideas, we are able to effectively scale the online defense algorithm to large instances of our dynamic security model. Defense policies were successfully computed for a graph consisting of 134 conditions (nodes), 143 exploits (hyperedges), 64 defense actions, and 30 security alerts (resulting in over 10^9 possible observation vectors). The resulting number of security states exceeded 100 million.

6.3.2. An Illustrative Example

We investigate an illustrative example of the defense problem using the sample dependency graph of Fig. 6.2. We assume $n_a = 3$ attacker types of varying aggression (described by their probabilities of attack and success, dictating the rate of movement through the system), knowledge (described by the separation between $\overline{\alpha}_{e_j}(\varphi_i)$ and $\underline{\alpha}_{e_j}(\varphi_i)$ terms in Eq. (6.2)), and stealthiness (described by the probabilities of detection and false alarm). Specifically, the three attack types $\Phi = \{\varphi_1, \varphi_2, \varphi_3\}$ capture the following behavior.

	aggression	knowledge	stealthiness
φ_1	low	low	low
φ_2	moderate	high	high
φ_3	high	moderate	moderate

The problem parameters that capture the above behavior are now defined. Probabilities of attack for each exploit under each attacker type $\varphi_i \in \Phi$ are

$$\begin{aligned}
\left(\overline{\alpha}_{e_j}(\varphi_1), \underline{\alpha}_{e_j}(\varphi_1)\right) &= (0.5, 0.5) \text{ for all } e_j \in \mathcal{E}_0 \\
\left(\overline{\alpha}_{e_j}(\varphi_1), \underline{\alpha}_{e_j}(\varphi_1)\right) &= (0.3, 0.3) \text{ for all } e_j \in \mathcal{E} \setminus \mathcal{E}_0 \\
\left(\overline{\alpha}_{e_j}(\varphi_2), \underline{\alpha}_{e_j}(\varphi_2)\right) &= (0.8, 0.1) \text{ for } e_j \in \mathcal{E}_0 \\
\left(\overline{\alpha}_{e_j}(\varphi_2), \underline{\alpha}_{e_j}(\varphi_2)\right) &= (0.7, 0.3) \text{ for } e_j \in \{e_4, e_5, e_{10}, e_{12}, e_{13}\} \\
\left(\overline{\alpha}_{e_j}(\varphi_2), \underline{\alpha}_{e_j}(\varphi_2)\right) &= (0.6, 0.4) \text{ for } e_j \in \{e_6, e_7, e_8, e_9\} \\
\left(\overline{\alpha}_{e_j}(\varphi_3), \underline{\alpha}_{e_j}(\varphi_3)\right) &= (0.7, 0.4) \text{ for } e_j \in \mathcal{E}_0 \\
\left(\overline{\alpha}_{e_j}(\varphi_3), \underline{\alpha}_{e_j}(\varphi_3)\right) &= (0.6, 0.4) \text{ for } e_j \in \{e_4, e_5, e_{10}, e_{12}, e_{13}\} \\
\left(\overline{\alpha}_{e_j}(\varphi_3), \underline{\alpha}_{e_j}(\varphi_3)\right) &= (0.6, 0.5) \text{ for } e_j \in \{e_6, e_7, e_8, e_9\}.
\end{aligned}$$

Notice the separation between $\overline{\alpha}_{e_j}(\varphi_i)$ and $\underline{\alpha}_{e_j}(\varphi_i)$ for attacker types φ_2 and φ_3, reflecting a higher level of assumed knowledge than type φ_1. Similarly, probabilities of success are

$$\begin{aligned}
\beta_{e_j}(\varphi_1) &= 0.5 \text{ for all } e_j \in \mathcal{E}_0 \\
\beta_{e_j}(\varphi_1) &= 0.4 \text{ for all } e_j \in \mathcal{E} \setminus \mathcal{E}_0 \\
\beta_{e_j}(\varphi_2) &= 0.6 \text{ for all } e_j \in \mathcal{E}_0 \\
\beta_{e_j}(\varphi_2) &= 0.5 \text{ for all } e_j \in \mathcal{E} \setminus \mathcal{E}_0 \\
\beta_{e_j}(\varphi_3) &= 0.7 \text{ for all } e_j \in \mathcal{E}_0 \\
\beta_{e_j}(\varphi_3) &= 0.6 \text{ for all } e_j \in \mathcal{E} \setminus \mathcal{E}_0.
\end{aligned}$$

Probabilities of detection are provided in Table 6.1. Notice that increased stealthiness is represented by a lower probability of detection. Lastly, the probability of false alarm for each alert z_i under each type is $\zeta_i(\varphi_1) = 0.4$, $\zeta_i(\varphi_2) = 0.5$, and $\zeta_i(\varphi_3) = 0.6$. The space

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}
z_1	0.8	0.1	0	0	0	0	0	0	0	0	0.4	0	0
	0.3	0.4	0	0	0	0	0	0	0	0	0.3	0	0
	0.5	0.6	0	0	0	0	0	0	0	0	0.3	0	0
z_2	0	0.6	0.8	0	0	0	0	0	0	0	0.6	0	0
	0	0	0.4	0	0	0	0	0	0	0	0.2	0	0
	0	0.4	0.5	0	0	0	0	0	0	0	0.5	0	0
z_3	0	0	0	0.5	0	0.6	0.1	0	0	0	0	0	0
	0	0	0	0.4	0	0.4	0	0	0	0	0	0	0
	0	0	0	0.5	0	0.5	0.4	0	0	0	0	0	0
z_4	0	0	0	0	0.7	0	0.7	0	0	0	0	0	0
	0	0	0	0	0.3	0	0.5	0	0	0	0	0	0
	0	0	0	0	0.4	0	0.6	0	0	0	0	0	0
z_5	0	0	0	0	0	0	0	0.7	0.6	0	0	0	0
	0	0	0	0	0	0	0	0.4	0.3	0	0	0	0
	0	0	0	0	0	0	0	0.5	0.4	0	0	0	0
z_6	0	0	0	0	0	0	0	0	0.4	0.7	0	0	0
	0	0	0	0	0	0	0	0	0.2	0.5	0	0	0
	0	0	0	0	0	0	0	0	0.3	0.6	0	0	0
z_7	0	0	0	0	0	0	0	0	0	0	0	0.7	0
	0	0	0	0	0	0	0	0	0	0	0	0.4	0
	0	0	0	0	0	0	0	0	0	0	0	0.6	0
z_8	0	0	0	0	0	0	0	0	0	0	0	0	0.8
	0	0	0	0	0	0	0	0	0	0	0	0	0.4
	0	0	0	0	0	0	0	0	0	0	0	0	0.6

Table 6.1: Table of probabilities of detection for each attacker type. Columns represent attempted exploits wheres rows represent the triggered alert. Each entry represents the probability of detection, for a given exploit e_i (column) and alert z_j (row), for each type (from top to bottom), $\delta_{ij}(\varphi_1)$, $\delta_{ij}(\varphi_2)$, and $\delta_{ij}(\varphi_3)$.

of defense actions is constructed as the powerset of a set of binary defense actions, that is $\mathcal{U} = \mathscr{P}(\{u^1, u^2, u^3, u^4\})$, resulting in a total of $|\mathcal{U}| = 2^4 = 16$ defense actions. Each binary defense action induces a set of blocked exploits, defined as $\mathcal{B}(u^1) = \{e_1, e_2, e_3, e_4\}$, $\mathcal{B}(u^2) = \{e_5, e_7, e_{11}\}$, $\mathcal{B}(u^3) = \{e_8, e_9, e_{10}\}$, and $\mathcal{B}(u^4) = \{e_{12}, e_{13}\}$. The set of exploits that a defense action $u_t \in \mathcal{U}$ blocks is equal to the union of the blocked exploits of the binary defense actions that it contains, that is, $\mathcal{B}(u_t) = \bigcup_{u^i \in u_t} \mathcal{B}(u^i)$. Security states are assigned a cost of 1 for each goal condition, $\mathcal{N}^g = \{c_{11}, c_{12}\}$, that is contained in the state. The cost of each binary defense action is $c_u(u^i) = 0.25$, for all $i \in \{1, 2, 3, 4\}$. The cost weight in

Eq. (6.8) is set to $w = 0.5$ and the discount factor is $\rho = 0.95$. There are $n_s = 215$ security states (computed offline) and $n_z = 8$ security alerts leading to $|\mathcal{Y}| = 2^8 = 256$ observation vectors. All simulations for the example use $n_k = 1200$ particles to approximate the belief. The problem is assumed to start from the empty (safe) security state $s_0 = \varnothing$. The defender is initially completely uncertain of the true attacker type, reflected by a uniform belief over all attacker types. A sample evolution of the defense problem is illustrated in Fig. 6.7.

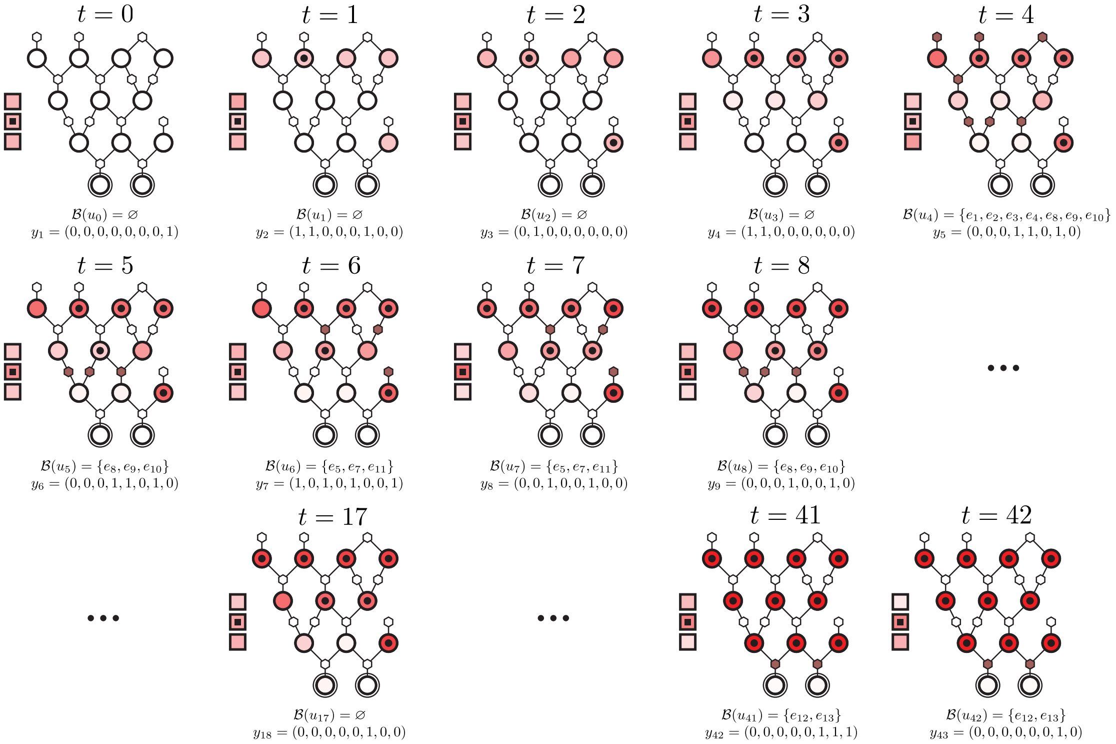

$t = 0$
$t = 1$
$t = 2$
$t = 3$
$t = 4$
$\mathcal{B}(u_0) = \varnothing$
$y_1 = (0,0,0,0,0,0,0,1)$
$\mathcal{B}(u_1) = \varnothing$
$y_2 = (1,1,0,0,0,1,0,0)$
$\mathcal{B}(u_2) = \varnothing$
$y_3 = (0,1,0,0,0,0,0,0)$
$\mathcal{B}(u_3) = \varnothing$
$y_4 = (1,1,0,0,0,0,0,0)$
$\mathcal{B}(u_4) = \{e_1, e_2, e_3, e_4, e_8, e_9, e_{10}\}$
$y_5 = (0,0,0,1,1,0,1,0)$
$t = 5$
$t = 6$
$t = 7$
$t = 8$
$\mathcal{B}(u_5) = \{e_8, e_9, e_{10}\}$
$y_6 = (0,0,0,1,1,0,1,0)$
$\mathcal{B}(u_6) = \{e_5, e_7, e_{11}\}$
$y_7 = (1,0,1,0,1,0,0,1)$
$\mathcal{B}(u_7) = \{e_5, e_7, e_{11}\}$
$y_8 = (0,0,1,0,0,1,0,0)$
$\mathcal{B}(u_8) = \{e_8, e_9, e_{10}\}$
$y_9 = (0,0,0,1,0,0,1,0)$
$t = 17$
$t = 41$
$t = 42$
$\mathcal{B}(u_{17}) = \varnothing$
$y_{18} = (0,0,0,0,0,1,0,0)$
$\mathcal{B}(u_{41}) = \{e_{12}, e_{13}\}$
$y_{42} = (0,0,0,0,0,1,1,1)$
$\mathcal{B}(u_{42}) = \{e_{12}, e_{13}\}$
$y_{43} = (0,0,0,0,0,0,1,0)$

Figure 6.7: Sample evolution of the defense problem. The current (true) security state s_t is represented by the tagged nodes. The true attacker type is represented by the tagged node in the panel to the left of each graph; the true type for the above simulation is φ_2. The defender's (marginal) probability for both the security state and true type is represented as a *heat-map* (representing values via colors), computed from the current belief state π_t, where a darker shade represents a higher probability. Blocked exploits $\mathcal{B}(u_t)$, represented by shaded hyperedges, and the observation vector y_{t+1} are displayed beneath each graph. The above sample evolution was performed using $n_{\text{sim}} = 5000$ simulations.

The computed defense policy is intuitive. Initially, in order to save on availability costs, the defense policy does not block any exploits. During this period of inaction, the defender's belief gradually assigns mass to worsening security states based on the received security alerts. The belief over the true attacker type (represented by the panel to the left of each graph in Fig. 6.7) is also updated as a function of the received alerts. Eventually the defense policy begins to deploy defense actions, blocking exploits that it believes are available to the attacker, as dictated by Eq. (6.1). The defense actions serve two purposes. First, the actions slow down the progression of the attacker through the system in the event that any of the blocked exploits are attempted. Second, the actions (along with the received observations) help the defender to gather information, serving to reduce its uncertainty of the true security state and attacker type. In order to lessen the negative impact to availability, the defense policy may prescribe the null action in some time-steps, as seen in $t = 17, 18$. In these cases, the defender will briefly wait for the attacker to progress before blocking exploits further downstream (as discussed at the end of Section 6.2.3, only blocking not yet successful exploits will impede the attacker's progression). This idle behavior only occurs in the early stages of the attack when the attacker is believed to be far from reaching a goal condition. When the defender's belief reflects that the attacker is close to reaching a goal condition,[††] the defense policy has no option but to block the exploit(s) that would allow the attacker to reach its goal(s), *e.g.* exploits e_{12} and e_{13} in time-steps $t = 40 - 42$ of Fig. 6.7. This defense action is persistent, resulting in the corresponding exploits being blocked for all subsequent time-steps. In summary, the defense policy initially behaves passively, placing priority on preserving availability, only deploying defense actions to slow the attacker and gain information. As the defender becomes more certain of the security state over time, it identifies and persistently blocks the exploits that would allow the attacker to

[††] In the event that the attacker has gained many conditions in a short period of time and the defender does not yet have a good estimate of the security state or attacker type, the defense policy will be more aggressive, blocking many exploits, until the defender's uncertainty is reduced.

reach a goal condition.

The defender's belief over the true attacker type exhibits more uncertainty than its belief over the security state. This is expected since the observed security alerts are more informative for the current progression of the attacker, *i.e.* the security state, than they are for inferring the true attacker type. In other words, the observed security alerts are largely consistent with the range of attacker behavior specified by the type space Φ. Nevertheless, the defender eventually becomes confident of the true attacker type (see time-steps $t = 40$ – 42 in Fig. 6.7), and even under the lack of complete certainty, is able to prescribe defense decisions that prevent the attacker from reaching the goal conditions.

The performance of the online defense algorithm improves as the number of simulation iterations n_{sim} increases, as shown by the plots in Fig. 6.8. For low simulation counts, *e.g.* $n_{\text{sim}} = 500$, the defense policy makes selections based on poor-quality estimates of the actions. This causes the defense policy to be overly aggressive initially, prescribing to block exploits from time-step $t = 0$ and unnecessarily restricting availability. Furthermore, due to the poor-quality estimates, the resulting defense policy also allows the attacker to reach a goal condition in many of the sample runs. As the number of simulations increases, more possible future histories are taken into account, resulting in higher quality estimates of actions and a better performing defense policy (as evidenced by the remaining plots in Fig. 6.8). The number of times that the attacker reaches a goal state decreases as the number of simulations increases. For $n_{\text{sim}} = 5000$, the attacker failed to reach any goal in all of the 20 sample runs.

6.3.3. A Remark on the Processing of Security Alerts

A particularly desirable feature of our state-based dynamic security model is in regard to how security alerts are processed, specifically false alarms. Existing approaches, such as the CSM (cooperating security managers) system of [White et al., 1996] or the EMERALD (event

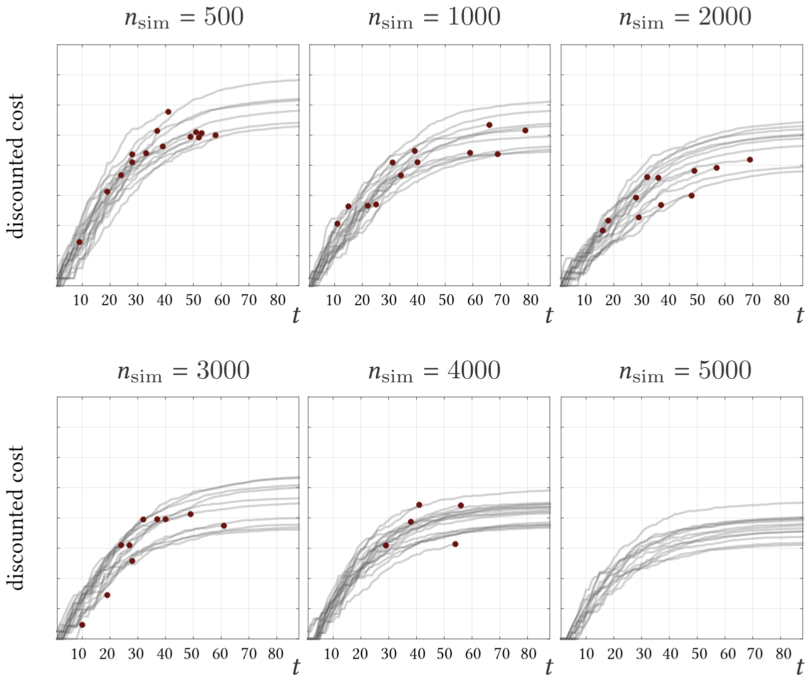


Figure 6.8: Discounted costs of sample paths for various simulation counts. The behavior of the defense policy (for the example system of Fig. 6.2) is demonstrated for simulation counts $n_{\text{sim}} = 500, 1000, 2000, 3000, 4000, 5000$. The simulations for each value of n_{sim} are initiated by randomly assigning a true attacker type uniformly from $\Phi = \{\varphi_1, \varphi_2, \varphi_3\}$. For each value of n_{sim}, the discounted cost is plotted (versus the time-step) for 20 sample paths. Trajectories that terminate in a marker represent sample paths where the attacker reached a goal state.

monitoring enabling response to anomalous live disturbances) system developed by [Porras & Neumann, 1997], attempt to deal with false alarms by defining *metrics* that reflect both the severity of an attack and the confidence that it is a real intrusion. In the presence of a high-rate of false alarms, these metric-based approaches can incorrectly classify benign security alerts as real intrusions. The state-based approach of our model avoids this drawback. Since the security state precisely describes what exploits are available to the attacker, via the set $\mathcal{E}(s)$ in Eq. (6.1), the defender is able to use the likelihood of the individual security states in its belief to weigh new security information. To see this, consider the following example:

Consider belief matrices $\pi, \pi' \in \Delta(\mathcal{S} \times \Phi)$ such that for some security state $s_i \in \mathcal{S}$ and type $\varphi_l \in \Phi$, $\pi^{il} > 0$ and $\pi'^{il} = 0$. Let there be a single available exploit in state s_i, that is, $\mathcal{E}(s_i) = \{e\}$ and assume that if exploit e is attempted, it generates the unique alert z (that is, no other exploit attempt can trigger alert z). If the defender possesses belief π and sees alert z, then the belief update allows for the possibility of the alert being generated by an attempt of exploit e. On the other hand, if π' is the current belief and alert z is received, the defender can say with certainty that the alert was a false alarm. In general, the likelihood of the individual security states in the current belief influence how security alert information is processed. In our simulations, we have observed that even in situations where the false-alarm rate is high, the defender is able to accurately track the true security state over time.

6.4. Conclusion

The complex nature of sophisticated cyber attacks necessitates the development of a defense system that is capable of prescribing defense actions in real-time that both mitigate the attack and preserve availability, all while enabling a solution that scales to realistically-sized cyber networks. Furthermore, the defense scheme must be able to operate under uncertainty of the attacker's strategy, the inherently noisy security alert information generated by the intrusion detection system, as well as use the evaluated effectiveness of previously deployed defense actions to influence future defense decisions. The state-based model introduced in this chapter addresses all of the above mentioned concerns. Specifically, using ideas from stochastic control theory, we can precisely model how the security status of the system evolves as a function of both the attacker's and defender's actions and formulate how the defender can use its imperfect information to specify *optimal* defense actions over time. Scalability is achieved via a sample-based, online defense algorithm that takes advantage of the structure of the security model to enable computation in large-scale domains.

Chapter 7

On Monotonicity Properties of Optimal Policies for POMDPs on Partially Ordered Spaces

Many decision problems possess state-spaces where not all states are comparable. For example, in security settings we cannot always say whether one security state is *safer* than another state. Similarly, the observation signals that we receive from the environment are also not comparable. In this chapter, we investigate such settings in the context of POMDPS and aim to derive conditions to ensure that the optimal policy is monotone in the belief. While an intuitive property, the result is non-trivial to show, requiring us to propose a new stochastic order and a corresponding class of order-preserving matrices.

7.1. Introduction

Partially observable Markov decision processes (POMDPs) model settings in which decisions must be made over time subject to imperfect information of the underlying status of the system. They have found applications in a multitude of practical settings including scheduling, optimal stopping, learning theory, threat and failure response, spoken dialogue systems, robot navigation, and many more. Unfortunately, obtaining a solution to the POMDP, that is, solving for an optimal decision rule (termed an *optimal policy*), is a computationally difficult process, particularly for the high-dimensional problems found in realistic decision environments.

Structural results for POMDPs investigate conditions under which optimal policies possess desirable properties. For example, one such structural result involves determining conditions under which the optimal policy is increasing in the information/belief state, termed a *monotone policy*. Establishing such structure not only simplifies the search for an optimal policy (often a set of numbers is sufficient for characterizing monotone policies; [Lovejoy, 1987]), but also provides insight into the problem, quantifying the relationship between optimal policy structure and the information pattern of the problem.

7.1.1. Literature Review

Questions concerning the structure of optimal policies are fundamental to decision analysis, spanning back to the seminal works of [Girshick & Rubin, 1952] and [Bellman, 1955]. Early work in the area, such as that of [Derman & Sacks, 1960] and [Derman, 1963], focused on completely observable settings with the goal of determining the optimal time to replace a system that is probabilistically degrading over time, so-called *replacement rules*. In particular, [Derman, 1963] studied replacement rules for a completely-observable problem on a totally-ordered state-space and derived "monotonicity-preserving" conditions on the transition matrix (*i.e.* increasing failure rate or IFR) ensuring that the optimal decision rule takes a *control-limit* form.

Investigating structural properties in problems of imperfect information represent a significant complication, primarily due to the requirement to (partially) order beliefs. The work of [Ross, 1971], one of the first to investigate such properties under imperfect information (in the context of POMDPs), largely avoids this requirement by considering a two-state core process, resulting in a total order among beliefs. In his work, Ross introduced an additional action, *inspect*, serving to reveal the true state of the system, and derived conditions that ensured the optimal policy takes an at-most-four-region (AM4R) structure.[*] [Albright, 1979] considered a two-state process similar to that of [Ross, 1971], but restricted attention to actions that transition the system to improved states, rather than reveal information. Instead, information is revealed to the decision-maker via a finite set of observations, generated probabilistically via an observation matrix as a function of the underlying state. Under monotonicity conditions on the transition matrix and reward functions, as well as the assumption that the observation matrix is totally positive of order 2 (TP_2), see [Karlin, 1968], the optimal policy is monotone in the belief. Albright illustrates the difficulties associated with considering more than two core states, demonstrating that one loses important monotonicity properties of the belief update when first-order stochastic dominance is used to order beliefs. Nevertheless, building upon the structural results of [Porteus, 1975], [White, 1979] managed to derive sufficient conditions to ensure that optimal replacement policies are monotone under first-order stochastic dominance (complementing the completely observable and unobservable cases studied in [White, 1980]). While a significant contribution to the field, White's conditions are fairly restrictive, requiring an upper bound on the discount factor in addition to monotonicity conditions on the model parameters. [Lovejoy, 1987] derived less strict conditions by ordering beliefs using the monotone likelihood ratio

[*] [Rosenfield, 1976a, Rosenfield, 1976b] also derived conditions to ensure the AM4R property under a slightly different paradigm in which the state consists of the pair (i, k), representing that it has been k timesteps since the state was known to be in state i.

order, a stronger partial order than first-order stochastic dominance. In his work, Lovejoy presents natural sufficient conditions (monotonicity conditions and TP_2 transition matrices) that ensure monotone optimal replacement policies, avoiding the requirement to bound the discount factor.

The strength of the partial order used to compare beliefs is intimately related to the restrictiveness of the conditions involved for establishing the structural result. The conditions of [White, 1979] involve comparing beliefs in a first-order stochastic dominance sense, resulting in more restrictive conditions than those obtained when beliefs are compared using the stronger monotone likelihood ratio of [Lovejoy, 1987]. The reason for this disparity arises directly from the fact that the monotone likelihood ratio, unlike first-order stochastic dominance, is preserved under conditioning on new information, as demonstrated in [Lovejoy, 1987]. This property illustrates that the monotone likelihood ratio order is a more fitting stochastic order than first-order stochastic dominance for problems of imperfect information.

7.1.2. Contribution

In this chapter, we extend the results of [Lovejoy, 1987] to problems where the underlying state-space is partially ordered. With the exception of [White, 1979], the majority of existing work considers settings where the underlying state-space is totally ordered. The motivation for considering a partially ordered state-space is largely a practical one; many problems have state-spaces where one cannot necessarily label every state as *better* or *worse* than other states. Our model also considers observations that are partially ordered, modeling the fact that the quality of signals received from the environment is not always comparable. Under this setting, we investigate a similar topic as that of [White, 1979, Lovejoy, 1987], namely the structure of optimal replacement policies. Specifically, we consider two actions, one that lets the system operate uninterrupted and another that transitions the system to the best state with certainty (a problem that is often referred to as the *machine replacement problem*) and investigate conditions that ensure the optimal replacement policy is monotone in the belief.

The model of this chapter is motivated by a stylized version of the dynamic security model of Chapter 6. Consider a setting in which the attacker is progressively moving through the cyber network toward its goal(s) and the defender, using noisy security alerts, constructs a belief of the attacker's progression and attempts to determine when to reset the system to the initial (empty) security state. Under the natural subset order, the space of security states is partially ordered. A primary objective of the present chapter is to investigate if optimal policies for problems of this type exhibit any structure.

Due to the partial ordering of the underlying state-space, the standard monotone likelihood ratio definition does not apply. We propose a generalized definition of the monotone likelihood ratio, termed the *generalized monotone likelihood ratio*, along with a class of matrices, termed *generalized totally positive of order 2*, that preserve this order. Our proposed stochastic order possesses many desirable properties, permitting natural sufficient conditions to guarantee monotone optimal policies. The conditions we obtain are qualitatively similar to those of [Lovejoy, 1987], with the addition of a condition (on the observation probabilities) directly arising from the fact that the state and observation spaces are only partially ordered.

7.2. The Partially Observable Sequential Decision Model

Consider a finite time-horizon of length T. At each time t, the state of the system takes on one of finitely many states from the set $\mathcal{S} = \{s_1, \ldots, s_n\}$, where s_1 is termed the *best* state and s_n the *worst* state. The controller has access to two actions, $\mathcal{U} = \{u_0, u_1\}$, where u_0 lets the system evolve uninterrupted and u_1 transitions the system to state s_1 with certainty. Actions are costly – for a given state-action pair, an instantaneous cost $c(s_t, u_t)$ is incurred.

Let $c(s_t)$ denote the cost at the terminal stage, $t = T$. Given the current state $s_t = s_i$ and current action $u_t = u$, the system evolves probabilistically as dictated by the conditional transition probability matrices P^u, $u \in \mathcal{U}$, with elements $p_{ij}^u = P(S_{t+1} = s_j \mid S_t = s_i, U_t = u)$. The controller does not observe the underlying state s_t perfectly, instead it receives an observation $y_k \in \mathcal{Y} = \{y_1, \ldots, y_m\}$, at each time t, as dictated by the conditional observation (emission) matrix R, with elements $r_{jk} = P(Y_{t+1} = y_k \mid S_{t+1} = s_j)$. Notice for our model that, without loss of generality, the conditional observation probabilities are assumed to be independent of the control action. For a given iteration, events unfold in the following order:

1) A control action, $u_t = u \in \mathcal{U}$, is specified.

2) A state-dependent cost, $c(s_t, u_t)$, is incurred.

3) The state transitions to $s_{t+1} \in \mathcal{S}$ as dictated by the transition probabilities $p_{ij}^u = P(S_{t+1} = s_j \mid S_t = s_i, U_t = u)$.

4) An observation $y_{t+1} \in \mathcal{Y}$ is received as dictated by the conditional observation probabilities $r_{jk} = P(Y_{t+1} = y_k \mid S_{t+1} = s_j)$.

The information available to the controller at time t, represented by the history of actions and observations (as well as the distribution π_0 over the initial state), denoted by $h_t = (\pi_0, u_0, y_1, u_1, y_2, \ldots, u_{t-1}, y_t)$, can be summarized by a probability mass function over the state-space $\mathcal{S}$, termed an *information state* or *belief* $\pi_t \in \Delta(\mathcal{S})$ ([Åström, 1965], [Kumar & Varaiya, 1986]). The i'th component of belief $\pi \in \Delta(\mathcal{S})$ is the conditional probability that the system is in state $s_i \in \mathcal{S}$ given a history of h_t, that is, $\pi_i = P(S_t = s_i \mid H_t = h_t)$. Given new information, consisting of the current action u_t and the observation y_{t+1}, the belief is updated according to the recursive function $\tau : \Delta(\mathcal{S}) \times \mathcal{U} \times \mathcal{Y} \to \Delta(\mathcal{S})$ as $\pi_{t+1} =$

$\tau(\pi_t, u_t, y_{t+1}) = (\tau_1(\pi_t, u_t, y_{t+1}), \ldots, \tau_n(\pi_t, u_t, y_{t+1}))$ where each $\tau_j(\pi, u, y_k)$ is given by

$$\tau_j(\pi, u, y_k) = \frac{\sum_{i=1}^n \pi_i p_{ij}^u r_{jk}}{\sigma(\pi, u, y_k)} \tag{7.1}$$

where

$$\sigma(\pi, u, y_k) = \sum_{i=1}^n \sum_{j=1}^n \pi_i p_{ij}^u r_{jk}. \tag{7.2}$$

For later convenience, define $\sigma(\pi, u) \in \Delta(\mathcal{Y})$ as a probability mass function consisting of elements $\sigma(\pi, u, y_k)$ over all $y_k \in \mathcal{Y}$ for a fixed (π, u), and $r_i \in \Delta(\mathcal{Y})$ as a probability mass function consisting of elements r_{ik} over all $y_k \in \mathcal{Y}$.

The objective of the controller is to specify a control action at each time in order to minimize the expected discounted cost over the time horizon, given by

$$\mathbb{E}\left[\sum_{t=1}^{T-1} \rho^t c(s_t, u_t) + \rho^T c(s_T)\right]$$

where $\rho \in (0, 1)$ is the discount factor. The rule designating this choice is termed a *control policy*, denoted by $g = (g_1, g_2, \ldots, g_T)$, where each g_t is a function mapping an element of the probability simplex over $\mathcal{S}$, denoted by $\Delta(\mathcal{S})$, to a control action in $\mathcal{U}$. The optimal control policy, denoted by g^*, is the control policy that achieves the minimum expected total discounted cost.

The optimal policy can be characterized by the *value function*. Following [Porteus, 1975], [White, 1979], and [Lovejoy, 1987], define the function $\eta : \Delta(\mathcal{S}) \times \mathcal{U} \times \mathscr{B}(\mathcal{S}) \to \mathbb{R}$ as

$$\eta(\pi, u, V) = \sum_{i=1}^n \pi_i c(s_i, u) + \rho \sum_{k=1}^m \sigma(\pi, u, y_k) V(\tau(\pi, u, y_k)) \tag{7.3}$$

where $\mathscr{B}(\mathcal{S})$ be the set of bounded, real functions on $\Delta(\mathcal{S})$. The value function, denoted by

V_t^*, maps each belief $\pi \in \Delta(\mathcal{S})$ to a value representing the best that one can do from the given belief. Using the definition of h, the value function at any time t is given by

$$V_t^*(\pi) = \min_{u \in \mathcal{U}} \left(\eta(\pi, u, V_{t+1}^*)\right) .$$

Similarly, the optimal control policy $g^* = (g_1^*, g_2^*, \ldots, g_T^*)$, dictating the optimal control action at each time t, is given by

$$g_t^*(\pi) = \operatorname*{argmin}_{u \in \mathcal{U}} \left(\eta(\pi, u, V_{t+1}^*)\right) . \tag{7.4}$$

It is assumed that the states in $\mathcal{S}$ are partially ordered by $\succcurlyeq$, forming the partially ordered set (poset) $(\mathcal{S}, \succcurlyeq)$. Two states $s, s' \in \mathcal{S}$ are said to be unorderable, denoted by $s \parallel s'$, with respect to the partial order $\succcurlyeq$ if neither $s \succcurlyeq s'$ nor $s' \succcurlyeq s$. Furthermore, it is assumed that the observation space $\mathcal{Y}$ is partially ordered by $\succcurlyeq_{\mathcal{Y}}$, forming the poset $(\mathcal{Y}, \succcurlyeq_{\mathcal{Y}})$, where unorderable observations $y, y' \in \mathcal{Y}$ are denoted by $y \parallel_{\mathcal{Y}} y'$. Lastly, assume that the action space $\mathcal{U}$ is totally ordered by $\geq$, such that $u_1 \geq u_0$. Without loss of generality, assume that states and observations are indexed according to their respective partial orders, that is, if $s_i \succcurlyeq s_j$ ($y_k \succcurlyeq_{\mathcal{Y}} y_l$) then we index s_i and s_j (y_k and y_l) such that $i \geq j$ ($k \geq l$).

7.3. Preliminary Definitions

The structural results we are interested in obtaining in this paper require one to be able to compare beliefs, that is, to say when one belief π is *larger* than another belief π'. This necessitates the use of stochastic orders. Two such stochastic orders that will be useful for later discussion are first-order stochastic dominance and the monotone likelihood ratio order, defined below.

Definition 7.3.1 (First-order Stochastic Dominance). *Given elements $\pi, \pi' \in \Delta(\mathcal{S})$, π is said* *to be greater than* π' *with respect to* first order stochastic dominance (FOSD), *written* $\pi \succcurlyeq_s \pi'$, *if* $\sum_{j \geq i} \pi_j \geq \sum_{j \geq i} \pi'_j$ *for all* $i = 1, \ldots, n$.

Definition 7.3.2 (Monotone Likelihood Ratio). *Given elements* $\pi, \pi' \in \Delta(\mathcal{S})$, π *is said to be greater than* π' *with respect to the* monotone likelihood ratio (MLR), *written* $\pi \succcurlyeq_r \pi'$, *if* $\pi_i \pi'_j \geq \pi_j \pi'_i$ *for every* $i \geq j$.

The monotone likelihood ratio is a stronger partial order than first-order stochastic dominance, in the sense that if $\pi \succcurlyeq_r \pi'$ then $\pi \succcurlyeq_s \pi'$ (shown in [Whitt, 1979]). The above definitions apply in the case where the underlying state-space $\mathcal{S}$ is totally ordered, that is, for any two $s_i, s_j \in \mathcal{S}$, one can write either $s_i \leq s_j$ or $s_i \geq s_j$. Since the state-space (and observation-space) is assumed to be partially ordered in our model, we cannot directly make use of the above definitions.

First-order stochastic dominance has been generalized to the case where the underlying space is partially ordered. Let I_K denote the indicator vector, containing a one for all elements in the set K and a zero otherwise. The definition below, which we refer to as *generalized first-order stochastic dominance* (GFOSD), is courtesy of [White, 1979].

Definition 7.3.3 (Generalized First-order Stochastic Dominance). *Given elements* $\pi, \pi' \in \Delta(\mathcal{S})$, π *is said to be greater than* π' *with respect to* generalized first order stochastic dominance (GFOSD), *written* $\pi \succcurlyeq_{gs} \pi'$, *if* $\pi I_K \geq \pi' I_K$ *for all* $K \in \mathcal{K} = \{K \subseteq S \mid s_i \in K, s_j \succcurlyeq s_i \implies s_j \in K\}$.

It is worth noting that GFOSD reduces to FOSD (Definition 7.3.1) in the case where the underlying space is totally ordered; the set $\mathcal{K}$ reduces to contain sets of the form $\{s_1, \ldots, s_n\}$,$\{s_2, \ldots, s_n\}, \ldots, \{s_n\}$ (since all states are comparable).

Useful characterizations exist for both FOSD and GFOSD. A common characterization for FOSD, courtesy of [Stoyan & Daley, 1983], is as follows: π is said dominate π' with respect to $\succcurlyeq_s$ if and only if $\sum_i \pi_i f(s_i) \geq \sum_i \pi'_i f(s_i)$ for all increasing functions $f : \mathcal{S} \to \mathbb{R}$.

An analogous characterization for GFOSD is courtesy of [Kamae et al., 1977]. Let us first define the notion of $\succcurlyeq$-increasing functions: a function $f : \mathcal{S} \to \mathbb{R}$ is said to be $\succcurlyeq$-increasing if for any $s_i, s_j \in \mathcal{S}$ such that $s_i \succcurlyeq s_j$ we have that $f(s_i) \geq f(s_j)$. The characterization of GFOSD, restated in terms of the notation of our paper, is summarized by the following lemma.

Lemma 7.3.1 ([Kamae et al., 1977]). *Given elements $\pi, \pi' \in \Delta(\mathcal{S})$, π is said to dominate π' with respect to $\succcurlyeq_{gs}$ if and only if $\sum_i \pi_i f(s_i) \geq \sum_i \pi'_i f(s_i)$ for all $\succcurlyeq$-increasing functions f.*

7.4. Generalization of the Monotone Likelihood Ratio Order to Partially Ordered Spaces

As mentioned earlier, unlike FOSD, the (stronger) MLR order survives conditioning upon new information (see Section 6 of [Lovejoy, 1987]). This property allows for more natural conditions to ensure monotone optimal policies. One issue is that the definition of MLR assumes that the underlying space is totally ordered, an assumption that does not hold in our model. As a result, we propose a generalized definition of the MLR order for the case where the underlying space is partially ordered.

Definition 7.4.1 (Generalized Monotone Likelihood Ratio). *Given elements $\pi, \pi' \in \Delta(\mathcal{S})$, π is said to be greater than π' with respect to the* generalized monotone likelihood ratio (GMLR), *written $\pi \succcurlyeq_{gr} \pi'$, if*

$$\begin{aligned} \pi_i \pi'_j &\geq \pi_j \pi'_i \quad \textit{for } s_i \succcurlyeq s_j \\ \pi_i \pi'_j &= \pi_j \pi'_i \quad \textit{for } s_i \parallel s_j. \end{aligned}$$

Notice that if $\mathcal{S}$ were totally ordered, there would be no $s_i, s_j \in \mathcal{S}$ such that $s_i \parallel s_j$, resulting in $\succcurlyeq_{gr}$ reducing to $\succcurlyeq_r$ (Definition 7.3.2). Furthermore, analogous to the totally ordered case where $\pi \succcurlyeq_r \pi'$ implies $\pi \succcurlyeq_s \pi'$, the GMLR order is stronger than GFOSD, a property which is formalized by the following lemma.

Lemma 7.4.1. *If $\pi \succcurlyeq_{gr} \pi'$ then $\pi \succcurlyeq_{gs} \pi'$.*

Proof. See Appendix D.1. □

An important step in establishing the desired threshold properties is characterizing the class of matrices that preserve the GMLR order, that is, given $\pi \succcurlyeq_{gr} \pi'$, finding the class of matrices P such that $\pi P \succcurlyeq_{gr} \pi' P$. In the case where the underlying space is totally ordered, it is known that the MLR order is preserved by a class of matrices termed *totally positive of order 2* (TP$_2$), that is, if $\pi \succcurlyeq_r \pi'$ and P is a stochastic, TP$_2$ matrix then $\pi P \succcurlyeq_r \pi' P$ (see [Karlin, 1968, Karlin & Rinott, 1980]). We define a generalized notion of TP$_2$ matrices (in Definition 7.4.2) for the case where the underlying space is partially ordered, which we term *generalized totally positive of order* 2 (GTP$_2$), and show (in Proposition 1) that stochastic matrices of this type are sufficient for preserving the GMLR order.

Definition 7.4.2 (Generalized Totally Positive of Order 2). *A matrix $P \in \mathbb{R}^{n \times n}$ is said to be* generalized totally positive of order 2 (GTP$_2$) *if for every $s_k \succcurlyeq s_l$*

$$\begin{aligned} p_{lj}p_{ki} - p_{kj}p_{li} &\geq 0 \quad \text{for } s_i \succcurlyeq s_j \\ p_{lj}p_{ki} - p_{kj}p_{li} &= 0 \quad \text{for } s_i \parallel s_j \end{aligned}$$

Proposition 1. *If $\pi \succcurlyeq_{gr} \pi'$ and P is a stochastic, GTP$_2$ matrix then $\pi P \succcurlyeq_{gr} \pi' P$.*

Proof. See Appendix D.2. □

7.5. Main Result: Sufficient Conditions for Optimal Threshold Policies

Establishing threshold properties of optimal policies involve deriving the appropriate conditions on the state dynamics, observation dynamics, and structure of the instantaneous and terminal cost functions. The main result, stated below in Theorem 7.5.1, provides sufficient conditions for optimal policies to be monotone in the belief with respect to the GMLR order.

Theorem 7.5.1. *If $\pi \succeq_{gr} \pi'$ and the following conditions hold*

(a) $c(s)$ *is increasing in* s *on* $(\mathcal{S}, \succeq)$

(b) $c(s,u)$ *is increasing in* s *on* $(\mathcal{S}, \succeq)$ *for each* $u \in \mathcal{U}$

(c) $c(s,u_1) - c(s,u_0)$ *is decreasing in* s *on* $(\mathcal{S}, \succeq)$

(d) P^u *is GTP*$_2$ *for each* $u \in \mathcal{U}$

(e) $r_{ik}r_{jl} = r_{jk}r_{il}$ *if either* $s_i \parallel s_j$ *and* $y_k \succeq_{\mathcal{Y}} y_l$*, or* $s_i \succeq s_j$ *and* $y_k \parallel_{\mathcal{Y}} y_l$

(f) $r_i \succeq_{gr} r_j$ *for all* $s_i \succeq s_j$ *in* $\mathcal{S}$

then $g_t^*(\pi) \geq g_t^*(\pi')$ *for all* t.

The remainder of Section 7.5 will be dedicated to proving the above theorem. The results proceed by demonstrating, in Section 7.5.1, that conditions (a), (b), and (d) – (f) ensure that the value functions are increasing in the belief with respect to the GMLR order, that is, the value functions are increasing on the poset $(\Delta(\mathcal{S}), \succeq_{gr})$. This result is formally stated in Lemma 7.5.4. Next, an additional condition, (c), on the instantaneous cost function (decreasing differences), along with a result from [Topkis, 1978], ensures that the optimal policy is also monotone on the poset $(\Delta(\mathcal{S}), \succeq_{gr})$. The section concludes in Section 7.5.2 with the proof of Theorem 7.5.1.

7.5.1. Monotonicity of the Value Functions

Establishing monotonicity of the value functions on the poset $(\Delta(\mathcal{S}), \succeq_{gr})$, that is, showing that $V_t^*(\pi) \geq V_t^*(\pi')$ for any $\pi \succeq_{gr} \pi'$, requires first establishing some properties of the information dynamics. Specifically, the lemmas below (Lemmas 7.5.1 and 7.5.2) characterize monotonicity properties of the belief update function τ in both the observation and the belief. Lemma 7.5.1 introduces an assumption on the observation process, in turn establishing equivalence between monotonicity of the belief update in y on the observation poset $(\mathcal{Y}, \succcurlyeq_{\mathcal{Y}})$, for a fixed belief and action, and monotonicity of the observation pmf's $r_i \in \Delta(\mathcal{Y})$ in s_i. Lemma 7.5.2 shows equivalence between monotonicity of the belief update in π on the poset $(\Delta(\mathcal{S}), \succcurlyeq_{gr})$, for a fixed action and observation, and preservation of the order between MLR-orderable beliefs. Lemmas 7.5.1 and 7.5.2 are the partially ordered analogues to Lemma 1.2, parts (1) and (2), of the totally ordered setting found in [Lovejoy, 1987].

Lemma 7.5.1. *Assume that $r_{ik}r_{jl} = r_{jk}r_{il}$ if either $s_i \parallel s_j$ in $\mathcal{S}$ and $y_k \succcurlyeq_{\mathcal{Y}} y_l$ in $\mathcal{Y}$, or $s_i \succcurlyeq s_j$ in $\mathcal{S}$ and $y_k \parallel_{\mathcal{Y}} y_l$ in $\mathcal{Y}$. Then for any $\pi \in \Delta(\mathcal{S})$ and $u \in \mathcal{U}$,*

$$\tau(\pi, u, y_k) \succcurlyeq_{gr} \tau(\pi, u, y_l)$$

for all $y_k \succcurlyeq_{\mathcal{Y}} y_l$ in $\mathcal{Y}$ if and only if $r_i \succcurlyeq_{gr} r_j$ for all $s_i \succcurlyeq s_j$ in $\mathcal{S}$.

Proof. See Appendix D.3. □

Lemma 7.5.2. *For any $u \in \mathcal{U}$ and $y_k \in \mathcal{Y}$,*

$$\tau(\pi, u, y_k) \succcurlyeq_{gr} \tau(\pi', u, y_k)$$

for all $\pi \succcurlyeq_{gr} \pi'$ in $\Delta(\mathcal{S})$ if and only if $\pi P \succcurlyeq_{gr} \pi' P$ for all $\pi \succcurlyeq_{gr} \pi'$ in $\Delta(\mathcal{S})$.

Proof. See Appendix D.4. □

Before showing monotonicity of the value function, the following result regarding stochastic ordering of the pmf's $\sigma(\pi, u) \in \Delta(\mathcal{Y}), \pi \in \Delta(\mathcal{S}), u \in \mathcal{U}$ will be useful. This result, shown in Lemma 7.5.3 below, follows from the conditions that P^u is GTP$_2$ for each $u \in \mathcal{U}$ and the pmf's $r_i \in \Delta(\mathcal{Y})$ are increasing on $(\Delta(\mathcal{Y}), \succcurlyeq_{gr})$ in s_i on $(\mathcal{S}, \succcurlyeq)$.

Lemma 7.5.3. *If $\pi \succcurlyeq_{gr} \pi'$ and the following conditions hold*

1. *P^u is GTP$_2$ for each $u \in \mathcal{U}$*

2. *$r_i \succcurlyeq_{gr} r_j$ for all $s_i \succcurlyeq s_j$ in $\mathcal{S}$*

then $\sigma(\pi, u) \succcurlyeq_{gs} \sigma(\pi', u)$ for each $u \in \mathcal{U}$.

Proof. See Appendix D.5. □

Using Lemmas 7.5.1 through 7.5.3 and imposing monotonicity conditions on the instantaneous and terminal cost functions enable one to show that the optimal value function is increasing on the poset $(\Delta(\mathcal{S}), \succcurlyeq_{gr})$.

Lemma 7.5.4. *Let $\pi \succcurlyeq_{gr} \pi'$ and assume the following conditions hold*

1. *$c(s)$ is increasing in s on $(\mathcal{S}, \succcurlyeq)$*

2. *$c(s, u)$ is increasing in s on $(\mathcal{S}, \succcurlyeq)$ for each $u \in \mathcal{U}$*

3. *P^u is GTP$_2$ for each $u \in \mathcal{U}$*

4. *$r_{ik}r_{jl} = r_{jk}r_{il}$ if either $s_i \parallel s_j$ and $y_k \succcurlyeq_{\mathcal{Y}} y_l$, or $s_i \succcurlyeq s_j$ and $y_k \parallel_{\mathcal{Y}} y_l$*

5. *$r_i \succcurlyeq_{gr} r_j$ for all $s_i \succcurlyeq s_j$ in $\mathcal{S}$*

then $V_t^*(\pi) \geq V_t^*(\pi')$ for all t.

Proof. See Appendix D.6. □

7.5.2. Proof of the Main Result

Recall the function $\eta : \Delta(\mathcal{S}) \times \mathcal{U} \times \mathscr{B}(\mathcal{S}) \to \mathbb{R}$ of Eq. (7.3) and consider Lemma 7.5.5 below, a special case of Lemma 6.1 from [Topkis, 1978], restated using the notation of our model.

Lemma 7.5.5 ([Topkis, 1978]). *If $\eta(\pi, u, V^*_{t+1})$ has decreasing differences in (π, u) on the space $(\Delta(\mathcal{S}), \succeq_{gr}) \times \mathcal{U}$, then there exists a function $g^*_t(\pi) = \operatorname{argmin}_{u \in \mathcal{U}} \left(\eta(\pi, u, V^*_{t+1})\right)$ that is nondecreasing in π on $(\Delta(\mathcal{S}), \succeq_{gr})$.*

Under condition (c) of Theorem 7.5.1, Lemma 7.5.5 allows us to translate monotonicity of the value function into monotonicity of optimal policies. The proof of Theorem 7.5.1 is now possible.

Proof of Theorem 7.5.1. First, we show that $\eta(\pi, u, V^*_{t+1})$ has decreasing differences in (π, u) on $(\Delta(\mathcal{S}), \succeq_{gr}) \times \mathcal{U}$, that is, $\eta(\pi, u_1, V^*_{t+1}) - \eta(\pi, u_0, V^*_{t+1})$ is decreasing on $(\Delta(\mathcal{S}), \succeq_{gr})$. Then, application of Lemma 7.5.5 proves the result. Recall that $\tau(\pi, u_1, y_k) = v_1$ for any $\pi \in \Delta(\mathcal{S})$, $y_k \in \mathcal{Y}$, that is, the reset action u_1 causes the system state to transition to s_1 with certainty. Using this fact, along with the definition of η, see Eq. (7.3), we can write the following

$$\begin{aligned}&\eta(\pi, u_1, V^*_{t+1}) - \eta(\pi, u_0, V^*_{t+1}) \\ &\quad = \sum_{i=1}^{n} \pi_i \left(c(s_i, u_1) - c(s_i, u_0)\right) + \rho\left(V^*_{t+1}(v_1) - \sum_{k=1}^{m} \sigma(\pi, u_0, y_k) V^*_{t+1}(\tau(\pi, u_0, y_k))\right).\end{aligned}$$

Thus, for $\pi \succeq_{gr} \pi'$, we wish to show

$$\begin{aligned}&\sum_{i=1}^{n} \pi_i \left(c(s_i, u_1) - c(s_i, u_0)\right) + \rho\left(V^*_{t+1}(v_1) - \sum_{k=1}^{m} \sigma(\pi, u_0, y_k) V^*_{t+1}(\tau(\pi, u_0, y_k))\right) \\ &\leq \sum_{i=1}^{n} \pi'_i \left(c(s_i, u_1) - c(s_i, u_0)\right) + \rho\left(V^*_{t+1}(v_1) - \sum_{k=1}^{m} \sigma(\pi', u_0, y_k) V^*_{t+1}(\tau(\pi', u_0, y_k))\right). \end{aligned} \tag{7.5}$$

By condition (c) of Theorem 7.5.1, $c(s_i, u_1) - c(s_i, u_0)$ is decreasing in s_i on $(\mathcal{S}, \succeq)$. Consequently, since $\pi \succeq_{gs} \pi'$, application of Lemma 7.3.1 ensures that $\sum_{i=1}^{n} \pi_i \left(c(s_i, u_1) - c(s_i, u_0)\right) \leq \sum_{i=1}^{n} \pi'_i \left(c(s_i, u_1) - c(s_i, u_0)\right)$. Now, to ensure the relationship in Eq. (7.5) holds, we need to show that

$$\sum_{v=1}^{m} \sigma(\pi', u_0, y_k) V_{t+1}^*(\tau(\pi', u_0, y_k)) \leq \sum_{k=1}^{m} \sigma(\pi, u_0, y_k) V_{t+1}^*(\tau(\pi, u_0, y_k)) \tag{7.6}$$

Eq. (7.6) follows directly from the arguments found in the proof of Lemma 7.5.4 (see the arguments for establising the inequalities in Eqs. (D.2) and (D.3)). Specifically, notice that by conditions (a), (b), and (d) – (f), we have that $V_{t+1}^*(\pi) \geq V_{t+1}^*(\pi')$ for $\pi \succeq_{gr} \pi'$ by Lemma 7.5.4. Furthermore, by conditions (d) – (f), and Lemmas 7.3.1, 7.5.1, and 7.5.3, monotonicity of the value function ensures that

$$\sum_{k=1}^{m} \sigma(\pi', u_0, y_k) V_{t+1}^*(\tau(\pi', u_0, y_k)) \leq \sum_{k=1}^{m} \sigma(\pi, u_0, y_k) V_{t+1}^*(\tau(\pi', u_0, y_k)). \tag{7.7}$$

Additionally, by condition (d), Lemma 7.5.2, and monotonicity of the value function, we have

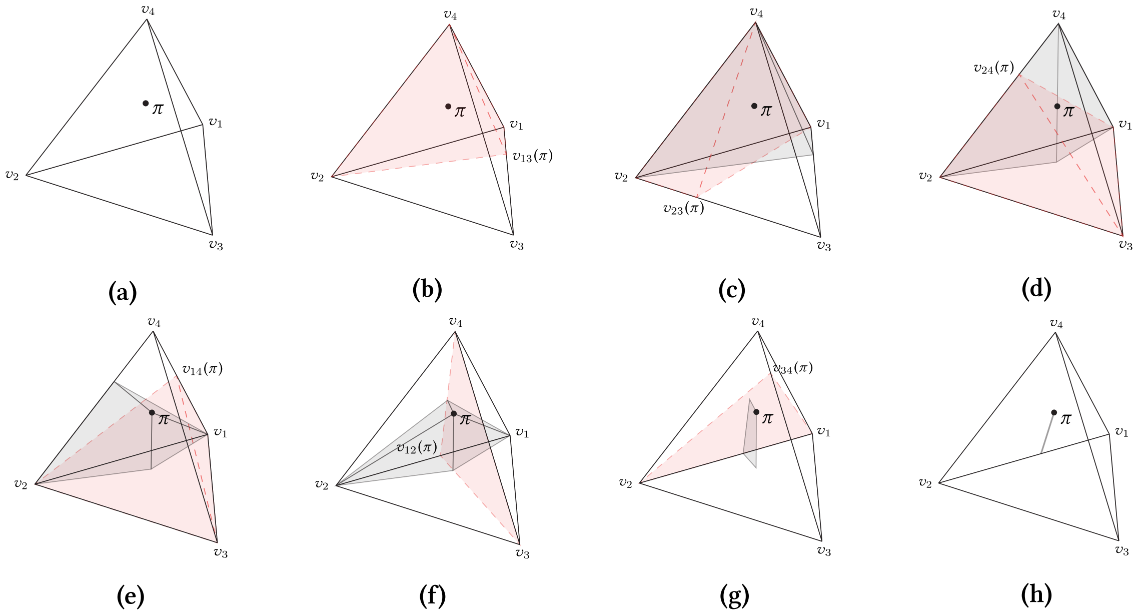
$$\sum_{k=1}^{m} \sigma(\pi, u, y_k) V_{t+1}^*(\tau(\pi', u, y_k)) \leq \sum_{k=1}^{m} \sigma(\pi, u, y_k) V_{t+1}^*(\tau(\pi, u, y_k)) \tag{7.8}$$

Eq. (7.6) follows by the transitivity of Eqs. (7.7) and (7.8), and thus $\eta(\pi, u, V_{t+1}^*)$ has decreasing differences in (π, u) on $(\Delta(\mathcal{S}), \succeq_{gr}) \times \mathcal{U}$. Application of Lemma 7.5.5 ensures that the optimal policy $g_t^*(\pi) = \operatorname{argmin}_{u \in \mathcal{U}} \left(\eta(\pi, u, V_{t+1}^*)\right)$ is increasing in π on $(\Delta(\mathcal{S}), \succeq_{gr})$. □

7.6. Visualizing the GMLR Order

The conditions for orderability under GMLR may raise questions as to the existence of orderable beliefs. For each pair of states, a halfspace in the probability simplex is induced if the states are orderable whereas a hyperplane is induced if the states are unorderable. In order to gain some intuition for the set of comparable beliefs under the GMLR order, it is useful to visualize the order for a given state-space ordering.

Consider a state-space consisting of four states $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$ with the ordering $s_3 \succcurlyeq s_1$, $s_3 \succcurlyeq s_2$, $s_4 \succcurlyeq s_2$, $s_4 \succcurlyeq s_1$, $s_1 \parallel s_2$, and $s_3 \parallel s_4$. For the given state-space ordering, Fig. 7.1 constructs the set of comparable beliefs π', for the given belief $\pi = (0.3, 0.2, 0.1, 0.4)$, such that $\pi \succcurlyeq_{gr} \pi'$.



(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)** **(h)**

Figure 7.1: Construction of orderable beliefs for a given state-space ordering. Halfspaces corresponding to orderable states are **(b)** $\pi_3\pi_1' \geq \pi_1\pi_3'$ arising from $s_3 \succcurlyeq s_1$ with intersect $v_{13}(\pi) = \left(1 - \frac{\pi_1}{\pi_1+\pi_3}, 0, \frac{\pi_1}{\pi_1+\pi_3}, 0\right)$, **(c)** $\pi_3\pi_2' \geq \pi_2\pi_3'$ from $s_3 \succcurlyeq s_2$ with $v_{23}(\pi) = \left(0, \frac{\pi_2}{\pi_2+\pi_3}, 1 - \frac{\pi_2}{\pi_2+\pi_3}, 0\right)$, **(d)** $\pi_4\pi_2' \geq \pi_2\pi_4'$ from $s_4 \succcurlyeq s_2$ with $v_{24}(\pi) = \left(0, \frac{\pi_2}{\pi_2+\pi_4}, 0, 1 - \frac{\pi_2}{\pi_2+\pi_4}\right)$, **(e)** $\pi_4\pi_1' \geq \pi_1\pi_4'$ from $s_4 \succcurlyeq s_1$ with $v_{14}(\pi) = \left(\frac{\pi_1}{\pi_1+\pi_4}, 0, 0, 1 - \frac{\pi_1}{\pi_1+\pi_4}\right)$. Hyperplanes corresponding to unorderable states are **(f)** $\pi_2\pi_1' = \pi_1\pi_2'$ from $s_1 \parallel s_2$ with $v_{12}(\pi) = \left(1 - \frac{\pi_2}{\pi_1+\pi_2}, \frac{\pi_2}{\pi_1+\pi_2}, 0, 0\right)$, and **(g)** $\pi_4\pi_3' = \pi_3\pi_4'$ from $s_3 \parallel s_4$ with $v_{34}(\pi) = \left(0, 0, \frac{\pi_3}{\pi_3+\pi_4}, 1 - \frac{\pi_3}{\pi_3+\pi_4}\right)$. The resulting set of comparable beliefs $\pi \succcurlyeq_{gr} \pi'$ is given by the line in **(h)**.

One can perhaps imagine a state-space ordering that results in no beliefs that are comparable to a given belief π. For instance, if there are many unorderable pairs of states, with each one inducing a hyperplane in the simplex, the resulting intersection of hyperplanes could result in the single belief π. We conjecture that if there are at most $k-1$ pairs of unorderable states, where k is the dimension of the probability simplex (e.g. $k = 3$ in the above example), then the resulting set of beliefs is non-trivial (in the sense that $\Delta(\mathcal{S}) \setminus \{\pi\}$ is nonempty).

7.7. Discussion & Conclusion

We have derived conditions to ensure monotone optimal policies in the case where the underlying state-space is partially ordered. While an intuitive property, establishing the optimality of monotone policies is non-trivial, primarily due to the requirement to select an appropriate partial order on the belief space. In this chapter, we have introduced a new partial order, termed the GMLR order, that is appropriate for comparing beliefs when not all of the underlying states are orderable. Furthermore, we have introduced a class of matrices, GTP_2, that preserve the GMLR order.

The conditions presented in our work are natural and are qualitatively similar to those of [Lovejoy, 1987]. Conditions (a) and (b) of Theorem 7.5.1 require that the instantaneous and terminal costs are increasing as the state degrades. Condition (c) states that the cost of doing nothing increases on $(\mathcal{S}, \succcurlyeq)$ more quickly than the cost of resetting. Condition (d) is with respect to the state dynamics, requiring that transitions to worse states are more likely as the state degrades. Condition (e), arising from the partial ordering of the state and observation spaces, is new and imposes conditions on the observation probabilities. While we do not have a clear intuition for this requirement, it can be interpreted as a type of (stochastic) *indifference* between alternatives (either observations or states) that we can't compare. Finally, condition (f) means that we are more likely to see worse signals from the environment as the underlying state degrades.

Chapter 8

Summary & Directions for Further Research

This dissertation has investigated three questions related to decision-making under uncertainty in cyber-physical systems. Specifically, the work has addressed the following questions: 1) In the context of power systems and electricity markets, how can one design algorithms that guide self-interested agents to a socially optimal and physically feasible outcome, subject to the fact that agents possess localized information of the system and react to local signals? 2) When a system is under attack from a malicious agent, what models are appropriate for performing real-time and scalable threat assessment and response selection when we only have partial information about the attacker's intent and capabilities? 3) Under what conditions do optimal policies of POMDPs possess desirable structure (specifically, monotonicity in the belief)? The discussion that follows reiterates the key points and contributions involved in answering each of these questions, as well as providing a critique of the results and suggestions for further research.

Decentralized Operation of Power Systems and Markets

The first question, addressed in Chapters 3 and 4, involves the development of models that capture the salient physical features of the electrical grid and algorithms that permit agents to reach a socially optimal outcome subject to their informational constraints. A defining feature of the proposed models is the ability to take into account power losses (via the modified DC approximation, see Section 2.2) without giving up desirable properties (namely convexity) that enable one to guarantee convergence. The algorithms that guide agents to a socially optimal outcome are iterative in nature and involve the exchange of messages. In Chapter 3, these messages are the operating point directly (specifically, the voltage phase angles). Using the operating point as messages allows agents to localize their externality effects, resulting in a completely decentralized algorithm that efficiently guides agents to the optimal outcome. In Chapter 4, a market operator sends price signals to the agents (DistCos, GenCos, and TransCos) who reply with their optimizers (for a given price, TransCos undergo a message-exchange process similar to that of Chapter 3). Through appropriate price updating, agents are guided to an outcome that maximizes their financial surpluses (competitive equilibrium) which is shown to be socially optimal.

The models of chapters 3 and 4 can be extended in various ways. While the agents are self-interested, they are not strategic, in the sense that they do not need to be incentivized to follow the rules of the algorithm (i.e. they do not try to *game the system*). Revisiting the design of these algorithms in the presence of strategic behavior represents an interesting and challenging research question. Additional extensions to the models include the consideration of a multi-period setting with temporal constraints (e.g. generation ramp limits, load shifting) and the consideration of stochastic generation and demand.

Dynamic Security of Cyber-Physical Systems

The second question, addressed in Chapter 6, involves the development of a formal, state-based sequential decision model. By embedding a state space on the dependency graph, the model is able to capture the complex nature of the attacker's progression. Furthermore, by considering multiple attacker types, the model describes the defender's uncertainty over the true strategy of the attacker. Using the received security alerts, the defender maintains a belief over both the capabilities/progression of the attacker (security state) and its strategy (attacker type). While realistic instances of the model can be very large, the use of a sampling-based approach avoids the state-space explosion problem and permits efficient computation of defense policies.

The nature of the computed defense policies hints at rules for secure system design. In particular, an interesting research direction is to investigate if it is possible to design a cyber network such that its dependency graph possesses properties that lead to efficient defense. For instance, secure systems should possess dependency graphs that have many layers of exploits between entry points (initial exploits) and critical system elements (goal conditions), requiring that the attacker perform many stages of exploits to reach its goal(s). Such dependency graphs should also have a small number of initial exploits, minimizing the number of entry points that the attacker can use to launch an attack. Furthermore, to ease selection of defense actions, it is desirable for dependency graphs to possess bottlenecks for the attacker, that is, many attack pathways that all pass through a small number of exploits. Such a property will allow for the defense policy to prescribe defense actions that effectively block many pathways while minimizing the negative impact to availability. In the context of cyber-physical systems, physical functionality of the system should be spread out across the dependency graph (*i.e.* one should not be able to exert system-wide control from a single computer). While this may decrease the functionality of the system for trusted users, it limits how much damage an attacker can inflict on the physical system from a given goal condition. Additionally, the defender's belief over the security state will be more informative for inferring likely future physical contingencies, permitting more effective defense of the system.

One can use the dynamic security model of Chapter 6 as a basis for addressing zero-day exploits. Recall that the model assumed knowledge of all exploits that the attacker could use to reach its goal(s). This assumption may not always hold, especially in the case of sophisticated modern-day attacks (such as Stuxnet, see Section 5.1.1). By allowing the defender to possess uncertainty over the structure of the dependency graph, zero-day exploits can be viewed as edges that the defender does not know exist. The problem of defending the system would then involve learning the structure of the underlying dependency graph in addition to selecting defense actions.

Structural Properties of Optimal Policies for POMDPs

The third question is addressed in Chapter 7 under the assumption that the underlying state space is only partially ordered. In this setting, a generalized version of the monotone likelihood ratio (termed the GMLR order, see Definition 7.4.1) and an associated class of order-preserving matrices (termed GTP$_2$, see Definition 7.4.2) are introduced. Conditions are derived that ensure monotone optimal policies, with respect to the GMLR order, in a two-action POMDP setting.

While the conditions of Theorem 7.5.1 are quite natural, applying them to the dynamic security model of Chapter 6 presents difficulties. To see this, recall the stylized version of the security model described in Section 7.1.2. In order to apply Theorem 7.5.1, one must first show that the transition matrices are GTP$_2$. The conditions on the security model's parameters (probabilities of attack and success) in order to satisfy the GTP$_2$ property turn out to be very strict due to the additional requirement that the matrix is upper-triangular (arising from the monotonicity assumption on the attacker's behavior, Section 7.2). The conditions on the observation dynamics are similarly difficult to satisfy in the context of the security model.

Nevertheless, it is straightforward to construct POMDP instances that satisfy the conditions of Theorem 7.5.1. In these settings, the results allow one to prune the space of optimal policies and can thus be useful for designing efficient policy search algorithms. If one can determine the optimal action for a sample belief then one say something about the optimal action in beliefs that are comparable to the sample belief. Repeating this process for multiple sample beliefs is informative for knowing the optimal action in *regions* of the probability simplex, aiding the search for an optimal policy.

To further increase the applicability of the results, it would be useful to derive analogous conditions for settings with more than two actions. This extension is non-trivial, primarily due to the difficulty of ensuring monotonicity properties of the dynamic programming recursion. Furthermore, in the case where not all actions are orderable, one would need to determine how to partially order the action space.

A Closing Remark

The frequency of breaches in recent months has demonstrated that security events are on the rise. As our societies become more interconnected and reliant upon technology, these security events will start to have more disruptive impacts on our lives. Fortunately, as of now, our society has not been subject to attacks that have impacted our critical infrastructure. Ensuring that this remains the case will require a persistent and coordinated effort from industry, government, and academia.

Appendix A

Appendix: A Decentralized Multi-Area Optimal Power Flow Algorithm with Power Losses

A.1. Benchmark of the modified DC approximation

We demonstrate the accuracy of the modified DC approximation by carrying out load flow analyses under the AC, DC, and modified DC approximations on thirteen test systems with network sizes ranging from 9 buses to 3120 buses.[*] The computational efficiency of each load flow analysis under the three approximations (AC, DC, and modified DC) is also compared. A Dell PowerEdge R815 equipped with four AMD Opteron 6174 processors (each 12-core, 2.2GHz, 12 x 512KB L2 cache, 12MB L3 cache) and 128GB of RAM was used as the computing platform for the simulations. Load flow analyses were carried-out in MATLAB.

[*] Case data obtained from IEEE and Matpower [Zimmerman et al., 2011].

To provide some perspective on the modified DC power flow approximation, we first recall the classical DC approximation. Like the modified DC approximation, the DC approximation sets all voltages to 1 p.u. but uses the small angle approximations $\cos(\theta_n - \theta_m) \approx 1$ and $\sin(\theta_n - \theta_m) \approx \theta_n - \theta_m$ resulting in a linear expression in the voltage angles, denoted by $\bar{g}(\theta_{nm}) = B_{nm}(\theta_n - \theta_m)$. The DC approximation is frequently chosen over the nonlinear AC equation, see Eq. (2.1), for a multitude of reasons [Stott et al., 2009]: it yields unique solutions, it lends itself to simple and efficient (non-iterative) solution methods, the computation of a solution requires minimal network data, and its linear nature fits well with market operation. On the other hand, the main concerns of the DC power flow approximation are the potential inaccuracy of the resulting phase angles and power flows as well as the fact that it ignores real power losses (this follows from the symmetry of the equation, $\bar{g}(\theta_{nm}) = -\bar{g}(\theta_{mn})$).

The modified DC approximation attempts to solve the concerns of the DC approximation while yielding a faster solution method than that of the AC power flow. The main value of modified DC approximation over the classical DC approximation comes from two factors: 1) the increased accuracy of the computed phase angles and resulting power flows; 2) the inclusion of real power losses. This improved approximation results from preserving some of the nonlinearity of the AC equation. Due to this nonlinearity, an iterative method (Newton's method) is needed in order to obtain a solution to the power flow equations (outlined below).

Load Flow – Modified DC Approximation

Step 0 – Initialization

Initialize: Set $t = 0$, $\boldsymbol{\theta}^0 = \mathbf{0}$ (flat start) and choose stopping threshold $\varepsilon > 0$.

Compute initial mismatch: $\Delta\mathbf{P}(\boldsymbol{\theta}^0) = \left(f_1(\boldsymbol{\theta}^0) - I_1, \ldots, f_{n_b}(\boldsymbol{\theta}^0) - I_{n_b}\right)$ where $f_n(\boldsymbol{\theta}) = \sum_{m \in \mathcal{N}_b} g(\theta_{nm})$ and I_n is the net real power injection at bus i.

Step 1 – Update

Update Jacobian: $[\mathbf{J}(\boldsymbol{\theta}^t)]_{ij} = \left[\frac{\partial \Delta \mathbf{P}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^t} \right]_{ij} = \begin{cases} \sum_{k \in \bar{\mathcal{R}}_i} (B_{ik} + G_{ik}\theta^t_{ik}) & \text{if } j = i \\ -B_{ij} - G_{ij}\theta^t_{ij} & \text{if } j \in \bar{\mathcal{R}}_i \\ 0 & \text{if } j \notin \bar{\mathcal{R}}_i \end{cases}$ where $\theta^t_{ik} = \theta^t_i - \theta^t_k$.

Compute mismatch: $\Delta \mathbf{P}(\boldsymbol{\theta}^t) = \left(f_1(\boldsymbol{\theta}^t) - I_1, \ldots, f_{n_b}(\boldsymbol{\theta}^t) - I_{n_b} \right)$

Update voltage angle: $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \mathbf{J}^{-1}(\boldsymbol{\theta}^t)\Delta \mathbf{P}(\boldsymbol{\theta}^t)$

Step 2 – Termination

If $||\Delta \mathbf{P}(\boldsymbol{\theta}^t)||_2 < \varepsilon$, terminate; else, increment counter, $t \leftarrow t + 1$, and return to step 1.

The accuracy and efficiency of the modified DC approximation when tested on the thirteen test systems is highlighted in Tables A.1 – A.3. Table A.1 presents the power flow and phase angle errors.[†] In all cases, the modified DC approximation yielded a lower total power flow error than the DC approximation (computed by summing, over all lines, the modulus of the power flow errors with respect to the AC load flow solution). Overall, the modified DC approximation resulted in significant gains in accuracy, especially for large systems. Power flow losses under the modified DC approximation are very close to those obtained in the AC load flow solution, as seen in Table A.2. In terms of efficiency, the modified DC approximation exhibited a speedup of about three times compared to the AC load flow, sometimes nearly reaching a four-fold speedup, as seen in Table A.3. The reason for this is clear; the modified DC approximation only considers real power flows and thus the Jacobian in the load flow algorithm contains one quarter of the number of elements compared to in the full AC load flow algorithm. As a result, it is less of a computational load to execute the load flow under the modified DC approximation.

As seen in the benchmark tests, the modified DC approximation offers a good tradeoff between speed (the approximation results in significant speed-up compared to the AC

[†] The power flow equations are not antisymmetric thus errors are presented for both the forward and reverse flow directions.

case (n_b)	forward direction flow error				reverse direction flow error				phase angle error			
	total		maximum		total		maximum		average		maximum	
	DC	mDC	DC	mDC	DC	mDC	DC	mDC	DC	mDC	DC	mDC
1 (9)	0.1406	0.0271	0.0571	0.0090	0.1471	0.0306	0.0571	0.0090	0.2022	0.0545	0.2895	0.1638
2 (14)	0.4351	0.2570	0.1258	0.0349	0.2804	0.2479	0.0733	0.0356	1.0445	0.2820	1.9125	0.8009
3 (24)	2.5344	0.5573	0.2428	0.0414	2.1817	0.5802	0.2498	0.0451	0.6810	0.2297	1.1066	0.4786
4 (30)	0.1240	0.0761	0.0222	0.0082	0.1072	0.0767	0.0219	0.0082	**0.1244**	**0.4434**	**0.5391**	**1.1684**
5 (39)	4.0218	1.2781	0.5215	0.0988	3.7075	1.2340	0.5215	0.0965	1.1375	0.1383	1.8915	0.5719
6 (57)	1.1947	0.6329	0.1044	0.0347	1.1234	0.6378	0.0947	0.0347	**0.7631**	**1.0832**	**1.6388**	**3.3553**
7 (118)	7.9987	0.7047	0.6945	0.0292	8.0815	0.6751	0.6945	0.0224	2.9766	0.1424	4.6334	0.5018
8 (300)	52.5097	15.0033	4.8961	0.8431	51.6257	14.8962	4.8961	0.8431	0.4002	0.0812	0.4966	0.1465
9 (2383)	64.4591	17.7324	1.4037	0.0994	64.1097	17.7421	0.9842	0.0849	0.4096	0.0287	0.1333	0.0375
10 (2737)	21.6381	9.6140	0.9376	0.0428	21.3160	9.5977	0.2723	0.0409	0.1507	0.0329	0.0282	0.0135
11 (2746)	41.2418	13.8965	2.0422	0.1322	40.5679	13.8564	0.5199	0.1353	0.1309	0.0166	0.0612	0.0208
12 (3012)	86.9045	28.4258	3.0869	0.1296	86.3474	28.4230	1.5671	0.1227	0.5782	0.0395	0.1887	0.0484
13 (3120)	65.7196	26.3657	3.1535	0.1754	64.9084	26.3742	0.6869	0.0989	0.7663	0.0731	0.0755	0.0285

Table A.1: Comparison of the total/maximum real power flow errors (pu) and average/maximum phase angle errors (deg) of both the DC and modified DC methods (referred to as mDC in the table for brevity), compared to the AC load flow solution. Power flow errors are given for both the forward and reverse flow directions (due to the asymmetric AC and modified DC flow equations). Bolded (red) text highlights cases where the modified DC approximation results in a less accurate solution than the DC approximation.

case (n_b)	total losses		relative error
	AC	mDC	
1 (9)	0.0571	0.0481	0.1020
2 (14)	0.1591	0.1560	0.0545
3 (24)	0.7212	0.6771	0.0665
4 (30)	0.0253	0.0214	0.2312
5 (39)	0.5215	0.4660	0.2711
6 (57)	0.3063	0.2876	0.0878
7 (118)	2.0373	1.9913	0.0577
8 (300)	4.9011	4.0561	0.2066
9 (2383)	7.1588	6.9676	0.0475
10 (2737)	1.7968	1.7589	0.0394
11 (2746)	3.9025	3.8170	0.0348
12 (3012)	7.5892	7.2372	0.0662
13 (3120)	6.1845	5.8714	0.0481

Table A.2: Total real power losses (pu) and relative error (infinity norm) of the loss vector obtained from the modified DC approximation compared to the AC solution.

case (n_b)	DC	AC	mDC	speedup
1 (9)	3.00×10^{-5}	1.64×10^{-3}	5.71×10^{-4}	2.879
2 (14)	3.60×10^{-5}	2.74×10^{-3}	8.52×10^{-4}	3.215
3 (24)	4.60×10^{-5}	4.36×10^{-3}	1.31×10^{-3}	3.321
4 (30)	5.40×10^{-5}	6.56×10^{-3}	1.74×10^{-3}	3.775
5 (39)	2.43×10^{-4}	8.14×10^{-3}	2.19×10^{-3}	3.725
6 (57)	3.02×10^{-4}	1.36×10^{-2}	3.40×10^{-3}	3.989
7 (118)	1.94×10^{-4}	2.66×10^{-2}	8.75×10^{-3}	3.042
8 (300)	6.14×10^{-4}	1.50×10^{-1}	4.03×10^{-2}	3.728
9 (2383)	3.23×10^{-1}	8.677	2.279	3.807
10 (2737)	4.55×10^{-1}	1.21×10^{1}	3.228	3.752
11 (2746)	4.47×10^{-1}	1.08×10^{1}	3.372	3.201
12 (3012)	1.328	1.37×10^{1}	3.532	3.887
13 (3120)	1.517	1.61×10^{1}	4.130	3.893

Table A.3: Execution time (seconds) of the DC, AC, and modified DC load flow methods. The speedup factor of the modified DC approximation, compared to the AC method, is also included.

method) and accuracy. The modified DC approximation solves many of the concerns of the DC approximation, such as accuracy of the phase angles and resulting flows and provides a very good approximation of power losses (even in large systems).

A.2. Proof of Lemma 3.3.1

Proof. First, consider a change of variables from angles, $\boldsymbol{\theta}_{\mathcal{R}^a}$, to differences of angles $\phi_{\mathcal{R}^a} = \mathcal{A}_a \boldsymbol{\theta}_{\mathcal{R}^a}$, where $\mathcal{A}_a \in \mathbb{R}^{(\sum_{n \in \mathcal{N}_b^a} |\bar{\mathcal{R}}_n|) \times (|\mathcal{R}^a|-1)}$. We define $\bar{\mathbf{A}}_a \in \mathbb{R}^{(\sum_{n \in \mathcal{N}_b^a} |\bar{\mathcal{R}}_n|) \times |\mathcal{R}^a|}$ as follows

$$\bar{\mathbf{A}}_a = \begin{bmatrix} \bar{\mathbf{A}}_a^1 \\ \bar{\mathbf{A}}_a^2 \\ \vdots \\ \bar{\mathbf{A}}_a^{n_b^a} \end{bmatrix}, \text{ where } \bar{\mathbf{A}}_a^n = \begin{bmatrix} — & v_n^\top - v_{[\bar{\mathcal{R}}_n]_1}^\top & — \\ — & v_n^\top - v_{[\bar{\mathcal{R}}_n]_2}^\top & — \\ & \vdots & \\ — & v_n^\top - v_{[\bar{\mathcal{R}}_n]_{|\bar{\mathcal{R}}_n|}}^\top & — \end{bmatrix},$$

where $v_n \in \mathbb{R}^{|\mathcal{R}^a|}$ is the standard basis vector (zeros with a one in element n). Matrix $\mathcal{A}_a$ is formed by removing the column of $\bar{\mathbf{A}}_a$ corresponding to the slack bus index in $\mathcal{R}^a \cap \mathcal{N}_b^s$ (since the slack angle is fixed).

We now prove a result concerning the rank of the matrices $\mathcal{A}_a$, in Lemma A.2.1, below.

Lemma A.2.1. *The matrix $\mathcal{A}_a$ has full rank, that is, rank($\mathcal{A}_a$) = $|\mathcal{R}^a| - 1$.*

Proof. Define $\mathbf{B}_a := \mathcal{A}_a^\top \mathcal{A}_a \in \mathbb{R}^{(|\mathcal{R}^a|-1)\times(|\mathcal{R}^a|-1)}$ as

$$[\mathbf{B}_a]_{ij} = \begin{cases} 2|\bar{\mathcal{R}}_i| & \text{if } j = i \\ -2 & \text{if } j \in \bar{\mathcal{R}}_i \\ 0 & \text{if } j \notin \bar{\mathcal{R}}_i \end{cases}.$$

Matrix $\mathbf{B}_a$ has some special structure. First, notice that $\mathbf{B}_a$ is diagonally dominant. Furthermore, since we have removed the column in $\bar{\mathbf{A}}_a$ corresponding to the slack bus, each bus i that is immediately connected to the slack bus corresponds to a row i which satisfies $\left|[\mathbf{B}_a]_{ii}\right| = [\mathbf{B}_a]_{ii} > \sum_j \left|[\mathbf{B}_a]_{ij}\right|$ (strict diagonal dominance). As a result, matrix $\mathbf{B}_a$ falls within the class of *irreducibly diagonally dominant* matrices, known to be non-singular (Theorem 6.2.27 of [Horn & Johnson, 1985]). By the rank relation $\text{rank}(\mathcal{A}_a) = \text{rank}(\mathbf{B}_a) = |\mathcal{R}^a| - 1$ (Theorem 5.5.4 of [Mirsky, 2012] (p.155)), $\mathcal{A}_a$ is full rank. □

Using the result of Lemma A.2.1 we proceed to complete the proof of the Lemma. Define the composition $C_a = F_a \circ \mathcal{A}_a$, so that $C_a(\boldsymbol{\theta}_{\mathcal{R}^a}) = F_a(\mathcal{A}_a\boldsymbol{\theta}_{\mathcal{R}^a}) = F_a(\phi_{\mathcal{R}^a})$. We first show the strong convexity of $F_a(\phi_{\mathcal{R}^a})$ in $\phi_{\mathcal{R}^a}$. We compute the Hessian of $F_a(\phi_{\mathcal{R}^a})$ as

$$\nabla^2_{\phi_{\mathcal{R}^a}} F_a(\phi_{\mathcal{R}^a}) := \mathbf{M}_a(\phi_{\mathcal{R}^a}) + \mathbf{D}_a(\phi_{\mathcal{R}^a}). \tag{A.1}$$

The matrix $\mathbf{M}_a(\phi_{\mathcal{R}^a}) = \text{diag}(\mathbf{M}_a^1(\phi_{\mathcal{R}_1}), \ldots, \mathbf{M}_a^{n_b^a}(\phi_{\mathcal{R}_{n_b^a}}))$ is block-diagonal and symmetric and $\mathbf{D}_a(\phi_{\mathcal{R}^a}) = \text{diag}(\mathbf{D}_a^1(\phi_{\mathcal{R}_1}), \ldots, \mathbf{D}_a^{n_b^a}(\phi_{\mathcal{R}_{n_b^a}}))$ is diagonal where $\phi_{\mathcal{R}_n}$ (a subvector of $\phi_{\mathcal{R}^a}$) is the neighboring angle differences with respect to bus n. Submatrices $\mathbf{M}_a^n(\phi_{\mathcal{R}_n})$ and $\mathbf{D}_a^n(\phi_{\mathcal{R}_i})$, $n \in \mathcal{N}_b^a$, are

$$\begin{aligned}
\mathbf{M}_a^n(\phi_{\mathcal{R}_n}) &:= c_n''\Big(\hat{f}_n(\phi_{\mathcal{R}_n})\Big)\mathbf{m}_a^n(\phi_{\mathcal{R}_n})\mathbf{m}_a^n(\phi_{\mathcal{R}_n})^\top \\
\mathbf{D}_a^n(\phi_{\mathcal{R}^a}) &:= c_n'\Big(\hat{f}_n(\phi_{\mathcal{R}_n})\Big)\sum_{m=1}^{|\bar{\mathcal{R}}_n|} G_{[\phi_{\mathcal{R}_n}]_m} v_m v_m^\top
\end{aligned}$$

where the column vector $\mathbf{m}_a^n(\phi_{\mathcal{R}_n})$ is defined as

$$\mathbf{m}_a^n(\phi_{\mathcal{R}_n}) := \Big(B_{[\phi_{\mathcal{R}_n}]_1} + G_{[\phi_{\mathcal{R}_n}]_1}[\phi_{\mathcal{R}_n}]_1, \ldots, B_{[\phi_{\mathcal{R}_n}]_{|\bar{\mathcal{R}}_n|}} + G_{[\phi_{\mathcal{R}_n}]_{|\bar{\mathcal{R}}_n|}}[\phi_{\mathcal{R}_n}]_{|\bar{\mathcal{R}}_n|}\Big) \in \mathbb{R}^{|\bar{\mathcal{R}}_n|}.$$

We use $\hat{f}_n : \mathbb{R}^{|\bar{\mathcal{R}}_n|} \to \mathbb{R}$ to denote the injected power as a function of bus n's neighboring angle differences and $B_{[\phi_{\mathcal{R}_n}]_m}, G_{[\phi_{\mathcal{R}_n}]_m} > 0$ to denote susceptance and conductance, respectively, of the line corresponding to the angle difference $[\phi_{\mathcal{R}_n}]_m$. Each $\mathbf{M}_a^n(\phi_{\mathcal{R}_n})$, $n \in \mathcal{N}_b^a$, is positive semi-definite since $c_n''\Big(\hat{f}_n(\phi_{\mathcal{R}_n})\Big) \geq 0$ by assumption 3, thus $\mathbf{M}_a(\phi_{\mathcal{R}^a})$ is positive semi-definite. Each $\mathbf{D}_a^n(\phi_{\mathcal{R}_n})$ is positive definite due to the fact that $G_{nm} > 0$ for all $\{n, m\} \in \mathcal{E}_l$ and $c_n'\Big(\hat{f}_n(\phi_{\mathcal{R}_n})\Big) > 0$ by assumption 3, therefore, $\mathbf{D}_a(\phi_{\mathcal{R}^a}) \succ \mathbf{0}$. Thus, by (A.1), $\nabla^2_{\phi_{\mathcal{R}^a}} F_a(\phi_{\mathcal{R}^a}) \succ \mathbf{0}$ on $\mathbf{\Phi}_{\mathcal{R}^a}$ and hence $F_a(\phi_{\mathcal{R}^a})$ is strongly convex in $\phi_{\mathcal{R}^a}$. Recall that $C_a = F_a \circ \mathcal{A}_a$. Using the strong convexity of $F_a(\phi_{\mathcal{R}^a})$ in $\phi_{\mathcal{R}^a}$ and the fact that $\mathcal{A}_a$ is full rank, we have, for all $\boldsymbol{\theta}_{\mathcal{R}^a} \in \mathbf{\Theta}_{\mathcal{R}^a}$,

$$\nabla^2_{\boldsymbol{\theta}_{\mathcal{R}^a}} C_a(\boldsymbol{\theta}_{\mathcal{R}^a}) = \mathcal{A}_a^\top \nabla^2_{\phi_{\mathcal{R}^a}} F_a(\phi_{\mathcal{R}^a}) \mathcal{A}_a \succ \mathbf{0}$$

and thus $C_a(\boldsymbol{\theta}_{\mathcal{R}^a})$ is strongly convex in $\boldsymbol{\theta}_{\mathcal{R}^a}$. □

Appendix B

Appendix: A Decentralized Mechanism for Computing Competitive Equilibria in Deregulated Electricity Markets

B.1. Proof of Lemma 4.5.1

Proof. Let $\mathbf{v} = \boldsymbol{\theta}^i$. The Hessian of $\Psi^i_{\mathcal{TC}}(\mathbf{v}, \boldsymbol{\lambda})$, see Eq. (4.5), with respect to $\mathbf{v}$ is given by $\nabla^2_{\mathbf{vv}} \Psi^i_{\mathcal{TC}}\big(\mathbf{v}, \boldsymbol{\lambda}\big) = -\frac{1}{2}\nabla^2_{\mathbf{vv}}\Big(\sum_{(n,m)\in\vec{\mathcal{E}}^i_l} (\lambda_n + \lambda_m)\tilde{g}(v_{nm})\Big)$, where the first order terms do not enter into the expression. Define $\iota = \mathcal{N}^i_{\mathcal{TC}}$ as the (ordered) set of bus indices of TransCo i

and define

$$A_{jk} = \begin{cases} -\sum_{\{j,l\}\in\mathcal{E}_l^i} (\lambda_{\iota_j} + \lambda_{\iota_l}) G_{\iota_j \iota_l} & \text{if } j = k \\ -(\lambda_{\iota_j} + \lambda_{\iota_k}) G_{\iota_j \iota_k} & \text{if } \{j,k\} \in \mathcal{E}_l^i \\ 0 & \text{if } \{j,k\} \notin \mathcal{E}_l^i \end{cases}$$

for each $j, k = 1, \dots, |\iota| = |\mathcal{N}_{\mathcal{TC}}^i|$. By assumption 1, there exists an index $s \in \iota$ that corresponds to a slack bus. The Hessian $\nabla^2 \Psi_{\mathcal{TC}}^i$ is defined as matrix A with the s^{th} row and column removed. Consequently, $\nabla^2 \Psi_{\mathcal{TC}}^i$ belongs to the class of *irreducibly diagonally dominant* matrices, known to be non-singular (see Theorem 6.2.27 of [Horn & Johnson, 1985]). To see this, note that the Hessian is diagonally dominant for all rows. Additionally, it is strictly diagonally dominant in rows that correspond to buses that are immediately connected to a slack bus. By assumption 3, the diagonal elements of the Hessian are negative and by Prop. 2.2.20 of [Cottle et al., 1992], $\nabla^2 \Psi_{\mathcal{TC}}^i \prec \mathbf{0}$. □

B.2. Proof of Lemma 4.5.2

Proof. The Hessian of the Lagrangian, denoted by $\nabla^2_{\mathbf{xx}} \mathcal{L}$, is a square, block-diagonal matrix of dimension $\sum_{i\in\mathcal{DC}} |\mathcal{N}_{\mathcal{DC}^e}^i| + \sum_{i\in\mathcal{GC}} |\mathcal{N}_{\mathcal{GC}}^i| + |\mathcal{N}_b \setminus \mathcal{N}_b^s|$. It consists of three blocks, $\mathbf{U}$, $\mathbf{C}$, and $\mathbf{M}$, where $\mathbf{U}$ and $\mathbf{C}$ are diagonal matrices consisting of elements $(u_n^i)''$ (corresponding to DistCo units) and $-(c_n^i)''$ (corresponding to GenCo units), respectively. By assumption 2, we have $\mathbf{U}, \mathbf{C} \prec \mathbf{0}$. Similar to the proof of Lemma 4.5.1, matrix $\mathbf{M}$ can be shown to be an irreducibly diagonally dominant matrix with a negative diagonal and thus, again by Prop. 2.2.20 of [Cottle et al., 1992], $\mathbf{M} \prec \mathbf{0}$. Since $\nabla^2_{\mathbf{xx}} \mathcal{L}$ is block-diagonal with each block negative definite, we conclude $\nabla^2_{\mathbf{xx}} \mathcal{L} \prec \mathbf{0}$. □

B.3. Proof of Theorem 4.5.1

Proof. Consider Problem (Q), defined as

$$\max_{\boldsymbol{\omega}\in\Omega\subseteq\mathbb{R}^W}\{G(\boldsymbol{\omega}):\mathbf{r}(\boldsymbol{\omega})=(r_1(\boldsymbol{\omega}),\ldots,r_M(\boldsymbol{\omega}))=\mathbf{0}\}\,. \tag{Q}$$

Also, consider the following definition.

Definition B.3.1 (Global $\boldsymbol{\omega}$-max, $\boldsymbol{\nu}$-min saddle point [Morgan, 2015]). *A point* $(\hat{\boldsymbol{\omega}},\hat{\boldsymbol{\nu}})$ *is a* global $\boldsymbol{\omega}$-max, $\boldsymbol{\nu}$-min saddle point *for the Lagrangian* $\mathcal{M}(\boldsymbol{\omega},\boldsymbol{\nu}) = G(\boldsymbol{\omega}) - \boldsymbol{\nu}^\top\mathbf{r}(\boldsymbol{\omega})$ *if and only if* $\mathcal{M}(\boldsymbol{\omega},\hat{\boldsymbol{\nu}}) \le \mathcal{M}(\hat{\boldsymbol{\omega}},\hat{\boldsymbol{\nu}}) \le \mathcal{M}(\hat{\boldsymbol{\omega}},\boldsymbol{\nu})\ \forall\,\boldsymbol{\omega}\in\Omega,\boldsymbol{\nu}\in\mathbb{R}^M$.

The proof proceeds in two steps: (i) We prove a general result demonstrating that if $(\hat{\boldsymbol{\omega}},\hat{\boldsymbol{\nu}})$ is a global $\boldsymbol{\omega}$-max, $\boldsymbol{\nu}$-min saddle point for the Lagrangian $\mathcal{M}$ then $\hat{\boldsymbol{\omega}}$ is the global optimum for the problem (Q) (similar to the proof found in [Morgan, 2015]); (ii) We show that the pricing process generates a global $\mathbf{x}$-max, $\boldsymbol{\lambda}$-min saddle point for the Lagrangian $\mathcal{L}$ of Problem (P).

Part (i): Assuming that $(\hat{\boldsymbol{\omega}},\hat{\boldsymbol{\nu}})$ is a global $\boldsymbol{\omega}$-max, $\boldsymbol{\nu}$-min saddle point, we have $\mathcal{M}(\hat{\boldsymbol{\omega}},\hat{\boldsymbol{\nu}}) \le \mathcal{M}(\hat{\boldsymbol{\omega}},\boldsymbol{\nu})$ for all $\boldsymbol{\nu}$. Thus

$$G(\hat{\boldsymbol{\omega}}) - \sum_{m=1}^{M}\hat{\boldsymbol{\nu}}_m r_m(\hat{\boldsymbol{\omega}}) \le G(\hat{\boldsymbol{\omega}}) - \sum_{m=1}^{M}\boldsymbol{\nu}_m r_m(\hat{\boldsymbol{\omega}}). \tag{B.1}$$

Let there exist an index m' such that $r_{m'}(\hat{\boldsymbol{\omega}}) > 0$, then we can choose $\nu_{m'} \gg 0$ such that Eq. (B.1) is violated. Similarly, let there exist an index m'' such that $r_{m''}(\hat{\boldsymbol{\omega}}) < 0$, we can violate Eq. (B.1) by choosing $\nu_{m''} \ll 0$. Thus $r_m(\hat{\boldsymbol{\omega}}) = 0$ for all m and therefore $\hat{\boldsymbol{\omega}}$ is feasible for Problem (Q).

Since $(\hat{\boldsymbol{\omega}},\hat{\boldsymbol{\nu}})$ is a global $\boldsymbol{\omega}$-max, $\boldsymbol{\nu}$-min saddle point, we also have $\mathcal{M}(\boldsymbol{\omega},\hat{\boldsymbol{\nu}}) \le \mathcal{M}(\hat{\boldsymbol{\omega}},\hat{\boldsymbol{\nu}})$ for all $\boldsymbol{\omega}\in\Omega$. Thus $G(\boldsymbol{\omega}) - \sum_{m=1}^{M}\hat{\boldsymbol{\nu}}_m r_m(\boldsymbol{\omega}) \le G(\hat{\boldsymbol{\omega}}) - \sum_{m=1}^{M}\hat{\boldsymbol{\nu}}_m r_m(\hat{\boldsymbol{\omega}})$. Since $\mathbf{r}(\boldsymbol{\omega}) = \mathbf{0}$ for every feasible $\boldsymbol{\omega}$, we have that $G(\boldsymbol{\omega}) \leq G(\hat{\boldsymbol{\omega}})$ everywhere on $\{\boldsymbol{\omega}|\mathbf{r}(\boldsymbol{\omega}) = \mathbf{0}, \boldsymbol{\omega} \in \Omega\}$ and thus $\hat{\boldsymbol{\omega}}$ is optimal for Problem (Q).

Part (ii): Let $\mathbf{x}^* = \mathbf{x}(\boldsymbol{\lambda}^*) = \operatorname{argmax}_{\mathbf{x}\in\mathbf{X}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^*)$, where $\boldsymbol{\lambda}^*$ is the converged price vector obtained from the pricing process (Eq. 7.1). The profile $\mathbf{x}^*$ is returned from the agents when provided with the price $\boldsymbol{\lambda}^*$ (via the optimizations in Section 4.5.1). By assumption 2 and Corollary 4.5.1, $\mathbf{x}^*$ is unique. Notice that $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is a competitive equilibrium by Def. 4.4.1. Also, due to the concavity of $\mathcal{L}$ (Lemma 4.5.2), $\phi(\boldsymbol{\lambda}^*) = \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) \geq \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^*)$ for all $\mathbf{x} \in \mathbf{X}$. We know that $\nabla\phi(\boldsymbol{\lambda})|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}^*} = \mathbf{0}$ and thus, by Eq. (4.8), $\mathbf{h}(\mathbf{x}(\boldsymbol{\lambda}^*)) = \mathbf{h}(\mathbf{x}^*) = \mathbf{0}$. Consequently, $\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = W(\mathbf{x}^*) - (\boldsymbol{\lambda}^*)^\top \mathbf{h}(\mathbf{x}^*) = W(\mathbf{x}^*) = W(\mathbf{x}^*) - \boldsymbol{\lambda}^\top \mathbf{h}(\mathbf{x}^*) = \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda})$ for all $\boldsymbol{\lambda}$. In summary $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^*) \leq \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda})$ for all $\mathbf{x} \in \mathbf{X}$, $\boldsymbol{\lambda}$ and thus $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is a global $\mathbf{x}$-max, $\boldsymbol{\lambda}$-min saddle point for the Lagrangian $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$.

From parts (i) and (ii), we conclude that the pricing process generates the pair $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ where $\mathbf{x}^*$ is a globally optimal solution to Problem (P). □

Appendix C

Appendix: A POMDP Approach to the Dynamic Defense of Large-Scale Cyber Physical Systems

C.1. Defender's Belief State Update

The timing diagram in Fig. C.1 will be useful for the arguments of this section.

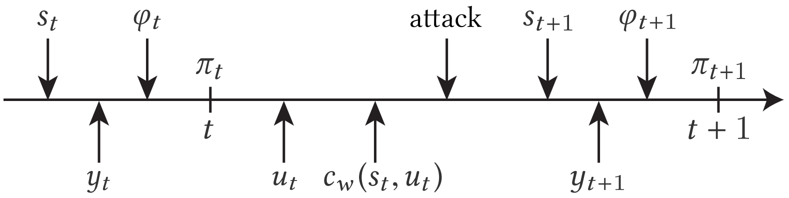


Figure C.1: Event and update timing for the dynamic security model.

The belief update of Eq. (6.4) is derived as follows

$$
\begin{aligned}
\pi_{t+1}^{jm} &= P(S_{t+1} = s_j, \Phi_{t+1} = \varphi_m \mid U_t = u, Y_{t+1} = y_k, \Pi_t = \pi_t) \\
&= \frac{\begin{array}{c} P(S_{t+1} = s_j, \Phi_{t+1} = \varphi_m \mid U_t = u, \Pi_t = \pi_t) \\ \cdot P(Y_{t+1} = y_k \mid S_{t+1} = s_j, U_t = u, \Pi_t = \pi_t) \end{array}}{P(Y_{t+1} = y_k \mid U_t = u, \Pi_t = \pi_t)} \\
&= \frac{p_{jm}^u(\pi_t) r_{jk}^u(\pi_t)}{\sigma(\pi_t, y_k, u)}
\end{aligned}
$$

The derivations for Eqs. (6.5) – (6.7) are now presented. Eq. (6.5) is obtained via

$$
\begin{aligned}
p_{jm}^u(\pi_t) &= P(S_{t+1} = s_j, \Phi_{t+1} = \varphi_m \mid U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} P(S_{t+1} = s_j, \Phi_{t+1} = \varphi_m, S_t = s_i, \Phi_t = \varphi_l \mid U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} P(S_{t+1} = s_j, \Phi_{t+1} = \varphi_m \mid S_t = s_i, \Phi_t = \varphi_l, U_t = u, \Pi_t = \pi_t) \\
&\qquad\qquad \cdot P(S_t = s_i, \Phi_t = \varphi_l \mid U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} P(S_{t+1} = s_j, \Phi_{t+1} = \varphi_m \mid S_t = s_i, \Phi_t = \varphi_l, U_t = u) \\
&\qquad\qquad \cdot P(S_t = s_i, \Phi_t = \varphi_l \mid \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} P(S_{t+1} = s_j \mid S_t = s_i, \Phi_t = \varphi_l, U_t = u) \\
&\qquad\qquad \cdot P(\Phi_{t+1} = \varphi_m \mid S_t = s_i, \Phi_t = \varphi_l, U_t = u) \\
&\qquad\qquad \cdot P(S_t = s_i, \Phi_t = \varphi_l \mid \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} \pi_t^{il} p_{ijl}^u q_{lm}.
\end{aligned}
$$

Eq. (6.6) is obtained via

$$\begin{aligned}
r^u_{jk}(\pi_t) &= P(Y_{t+1} = y_k \mid S_{t+1} = s_j, U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} P(Y_{t+1} = y_k, S_t = s_i, \Phi_t = \varphi_l \mid S_{t+1} = s_j, U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} P(Y_{t+1} = y_k \mid S_{t+1} = s_j, S_t = s_i, \Phi_t = \varphi_l, U_t = u, \Pi_t = \pi_t) \\
&\qquad\qquad \cdot P(S_t = s_i, \Phi_t = \varphi_l \mid U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} P(Y_{t+1} = y_k \mid S_{t+1} = s_j, S_t = s_i, \Phi_t = \varphi_l, U_t = u) \\
&\qquad\qquad \cdot P(S_t = s_i, \Phi_t = \varphi_l \mid \Pi_t = \pi_t) \\
&= \sum_{s_i \in \mathcal{S}, \varphi_l \in \Phi} \pi^{il}_t r^u_{ijkl}.
\end{aligned}$$

Finally, Eq (6.7) is derived as follows

$$\begin{aligned}
\sigma(\pi_t, y_k, u) &= P(Y_{t+1} = y_k \mid U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_j \in \mathcal{S}, \varphi_m \in \Phi} P(Y_{t+1} = y_k, S_{t+1} = s_j, \Phi_{t+1} = \varphi_m \mid U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_j \in \mathcal{S}, \varphi_m \in \Phi} P(Y_{t+1} = y_k \mid S_{t+1} = s_j, \Phi_{t+1} = \varphi_m, U_t = u, \Pi_t = \pi_t) \\
&\qquad\qquad \cdot P(S_{t+1} = s_j, \Phi_{t+1} = \varphi_m \mid U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_j \in \mathcal{S}, \varphi_m \in \Phi} P(Y_{t+1} = y_k \mid S_{t+1} = s_j, U_t = u, \Pi_t = \pi_t) \\
&\qquad\qquad \cdot P(S_{t+1} = s_j, \Phi_{t+1} = \varphi_m \mid U_t = u, \Pi_t = \pi_t) \\
&= \sum_{s_j \in \mathcal{S}, \varphi_m \in \Phi} r^u_{jk}(\pi_t) p^u_{jm}(\pi_t).
\end{aligned}$$

In order to define the transition probability, p^u_{ijl}, consider the set of *transition events*, denoted by $\mathcal{F}(s_i, s_j, \varphi_l, u)$, denoting the set of exploit events that could have caused the transition between s_i and s_j under action u and type φ_l. Each event in $v \in \mathcal{F}(s_i, s_j, \varphi_l, u)$

is a binary assignment (either successful or not successful) to each of the available exploits that are not blocked by the current defense action, $\mathcal{E}(s_i) \setminus \mathcal{B}(u)$. The transition probability is

$$p^u_{ijl} = \sum_{v \in \mathcal{F}(s_i, s_j, \varphi_l, u)} \left(\prod_{e_m \in v_1} \alpha_{e_m}(\varphi_l) \beta_{e_m}(\varphi_l) \cdot \prod_{e_m \in v_0} \left(1 - \alpha_{e_m}(\varphi_l) \beta_{e_m}(\varphi_l)\right) \right) \tag{C.1}$$

where we have used the fact that the events in $\mathcal{F}(s_i, s_j, \varphi_l, u)$ are disjoint. The set v_1 (resp. v_0) denotes the collection of exploits in v that must succeed (resp. must not succeed).

The observation probability r^u_{ijkl} is now defined. Introducing a variable E_t representing the set of exploits attempted by the attacker from state S_t, the probability r^u_{ijkl} is

$$\begin{aligned} r^u_{ijkl} &= P(Y_{t+1} = y_k \mid S_{t+1} = s_j, S_t = s_i, \Phi_t = \varphi_l, U_t = u) \\ &= \sum_{\mathcal{E}_a \in \mathscr{P}(\mathcal{E}(s_i))} P(Y_{t+1} = y_m, E_t = \mathcal{E}_a \mid S_{t+1} = s_j, S_t = s_i, \Phi_t = \varphi_l, U_t = u) \\ &= \sum_{\mathcal{E}_a \in \mathscr{P}(\mathcal{E}(s_i))} P(Y_{t+1} = y_m \mid E_t = \mathcal{E}_a, S_{t+1} = s_j, S_t = s_i, \Phi_t = \varphi_l, U_t = u) \\ &\qquad\qquad \cdot P(E_t = \mathcal{E}_a \mid S_{t+1} = s_j, S_t = s_i, \Phi_t = \varphi_l, U_t = u) \\ &= \sum_{\mathcal{E}_a \in \mathscr{P}(\mathcal{E}(s_i))} P(Y_{t+1} = y_m \mid E_t = \mathcal{E}_a, \Phi_t = \varphi_l) \\ &\qquad\qquad \cdot P(E_t = \mathcal{E}_a \mid S_{t+1} = s_j, S_t = s_i, \Phi_t = \varphi_l, U_t = u) \end{aligned} \tag{C.2}$$

where we have used the fact that the event $\{Y_{t+1} = y_m\}$ is independent of the event $\{S_{t+1} = s_j, S_t = s_i, U_t = u\}$ given the exploit attempt event $\{E_t = \mathcal{E}_a\}$. The probability of seeing a given observation vector given a set of exploit attempts, $P(Y_{t+1} = y_m \mid E_t = \mathcal{E}_a, \Phi_t = \varphi_l)$, is defined as

$$P(Y_{t+1} = y_m \mid E_t = \mathcal{E}_a, \Phi_t = \varphi_l) = \prod_{j \in \mathcal{A}} P(Y^j_{t+1} = y^j_m \mid E_t = \mathcal{E}_a, \Phi_t = \varphi_l)$$

where separability of the above terms follows from the fact that the elements of the observation vector are conditionally independent given the exploit attempt. Defining $\mathcal{E}(z_j)$ as the set of exploits that can trigger alert z_j, that is, $\mathcal{E}(z_j) = \{e_i \in \mathcal{E} \mid z_j \in \mathcal{Z}(e_i)\}$, each term in the above product is

$$P(Y_{t+1}^j = y_m^j \mid E_t = \mathcal{E}_a) = \begin{cases} \big(1 - \zeta_j(\varphi_l)\big) \prod\limits_{e_i \in \mathcal{E}_a \cap \mathcal{E}(z_j)} \big(1 - \delta_{ij}(\varphi_l)\big) & \text{if } y_m^j = 0 \\ 1 - \big(1 - \zeta_j(\varphi_l)\big) \prod\limits_{e_i \in \mathcal{E}_a \cap \mathcal{E}(z_j)} \big(1 - \delta_{ij}(\varphi_l)\big) & \text{if } y_m^j = 1 \end{cases}.$$

The probability of exploit attempts given a transition from s_i to s_j under action u and type φ_l, $P(E_t = \mathcal{E}_a \mid S_{t+1} = s_j, S_t = s_i, \Phi_t = \varphi_l, U_t = u)$, is

$$\begin{aligned} &P(E_t = \mathcal{E}_a \mid S_{t+1} = s_j, S_t = s_i, \Phi_t = \varphi_l, U_t = u) \\ &\quad = \frac{P(S_{t+1} = s_j \mid E_t = \mathcal{E}_a, S_t = s_i, \Phi_t = \varphi_l, U_t = u) P(E_t = \mathcal{E}_a \mid S_t = s_i, \Phi_t = \varphi_l)}{P(S_{t+1} = s_j \mid S_t = s_i, \Phi_t = \varphi_l, U_t = u)}. \end{aligned}$$

To define the probability $P(S_{t+1} = s_j \mid E_t = \mathcal{E}_a, S_t = s_i, \Phi_t = \varphi_l, U_t = u)$, let the set $\mathcal{F}(s_i, s_j, \varphi_l, u, \mathcal{E}_a)$ denote the collection of attempted exploit events that could have resulted in a transition to state s_j given that exploits $\mathcal{E}_a$ were attempted in state s_i under action u and type φ_l. Each event $v \in \mathcal{F}(s_i, s_j, \varphi_l, u, \mathcal{E}_a)$ is a binary assignment (either successful or not successful) to each of the available exploits that are attempted and not currently blocked, $(\mathcal{E}_a \cap \mathcal{E}(s_i)) \setminus \mathcal{B}(u)$. The probability is then given by

$$\begin{aligned} &P(S_{t+1} = s_j \mid E_t = \mathcal{E}_a, S_t = s_i, \Phi_t = \varphi_l, U_t = u) \\ &\quad = \sum_{v \in \mathcal{F}(s_i, s_j, \varphi_l, u, \mathcal{E}_a)} \left(\prod_{e_l \in v_1} \beta_{e_l}(\varphi_l) \cdot \prod_{e_l \in v_0} \big(1 - \beta_{e_l}(\varphi_l)\big) \right). \end{aligned}$$

The probability of exploits $\mathcal{E}_a$ being attempted given the current security state s_i, $P(E_t =$

$\mathcal{E}_a \mid S_t = s_i, \Phi_t = \varphi_l)$, is

$$P(E_t = \mathcal{E}_a \mid S_t = s_i, \Phi_t = \varphi_l) = \prod_{e_l \in \mathcal{E}_a \cap \mathcal{E}(s_i)} \alpha_{e_l}(\varphi_l) \cdot \prod_{e_l \in \mathcal{E}(s_i) \setminus \mathcal{E}_a} \left(1 - \alpha_{e_l}(\varphi_l)\right)$$

and $P(S_{t+1} = s_j \mid S_t = s_i, \Phi_t = \varphi_l, U_t = u)$ is the transition probability given by p^u_{ijl}.

Appendix D

Appendix: On Monotonicity Properties of Optimal Policies for POMDPs on Partially Ordered Spaces

D.1. Proof of Lemma 7.4.1

Proof. Let $\pi \succcurlyeq_{gr} \pi'$, so $\pi_i\pi'_j \geq \pi_j\pi'_i$ if $s_i \succcurlyeq s_j$ and $\pi_i\pi'_j = \pi_j\pi'_i$ if $s_i \parallel s_j$. Recall the definition of generalized first-order stochastic dominance (Definition 7.3.3). For each $K \in \mathcal{K}$, define $\bar{K} = \mathcal{S} \setminus K$. As a result of the definition of the set K, and the fact that $\pi \succcurlyeq_{gr} \pi'$, for each $(i, j) \in K \times \bar{K}$ there exists either an expression $\pi_i\pi'_j \geq \pi_j\pi'_i$ if $s_i \succcurlyeq s_j$ or $\pi_i\pi'_j = \pi_j\pi'_i$ if $s_i \parallel s_j$. For a given $K, \bar{K}$ pair, sum the corresponding expressions over all $(i, j) \in K \times \bar{K}$, yielding

$$\sum_{(i,j)\in K\times\bar{K}} \pi_i\pi'_j \geq \sum_{(i,j)\in K\times\bar{K}} \pi_j\pi'_i$$

due to the fact that $\pi \succcurlyeq_{gr} \pi'$. The above inequality can be factored into the form $\pi I_K \pi' I_{\bar{K}}$ $\geq \pi I_{\bar{K}} \pi' I_K$. Now,

$$\begin{aligned} \pi I_K \pi' I_{\bar{K}} &\geq \pi I_{\bar{K}} \pi' I_K \\ \equiv (\pi I_K)(1 - \pi' I_K) &\geq (1 - \pi I_K)(\pi' I_K) \\ \equiv \pi I_K - \pi I_K \pi' I_K &\geq \pi' I_K - \pi I_K \pi' I_K \\ \equiv \pi I_K &\geq \pi' I_K \end{aligned}$$

for each $K \in \mathcal{K}$, thus $\pi \succcurlyeq_{gs} \pi'$. □

D.2. Proof of Proposition 1

Proof. Let P be GTP$_2$ and $\pi \succcurlyeq_{gr} \pi'$. Denoting $P_{\circ,i}$ as the i'th column of matrix P, we wish to show that $\pi P_{\circ,i} \pi' P_{\circ,j} \geq \pi P_{\circ,j} \pi' P_{\circ,i}$ for $s_i \geq s_j$ and $\pi P_{\circ,i} \pi' P_{\circ,j} = \pi P_{\circ,j} \pi' P_{\circ,i}$ for $s_i \parallel s_j$. Equivalently, defining $q_{ij}(\pi, \pi') = \pi P_{\circ,i} \pi' P_{\circ,j} - \pi P_{\circ,j} \pi' P_{\circ,i}$, we wish to show that

$$\begin{aligned} &q_{ij}(\pi, \pi') \geq 0 \quad \text{for } s_i \geq s_j \\ &q_{ij}(\pi, \pi') = 0 \quad \text{for } s_i \parallel s_j. \end{aligned}$$

Observe that

$$\begin{aligned} q_{ij}(\pi, \pi') &= \pi P_{\circ,i} \pi' P_{\circ,j} - \pi P_{\circ,j} \pi' P_{\circ,i} \\ &= \pi (P_{\circ,i} P_{\circ,j}^\top - P_{\circ,j} P_{\circ,i}^\top) \pi'^\top. \end{aligned}$$

Define $A^{ij} = P_{\circ,i} P_{\circ,j}^\top - P_{\circ,j} P_{\circ,i}^\top$ and notice that A^{ij} is skew-symmetric, that is, $(A^{ij})^\top = -A^{ij}$. The (k,l)'th element of matrix A^{ij}, denoted by a_{kl}^{ij}, is given by $a_{kl}^{ij} = p_{lj} p_{ki} - p_{kj} p_{li}$ where

$a^{ij}_{kl} = 0$ for $k = l$. The function $q_{ij}(\pi, \pi') = \pi A^{ij} \pi'^{\top}$ can then be written as

$$\pi A^{ij} \pi'^{\top} = \sum_{l=1}^{n} \sum_{k=l+1}^{n} (p_{lj} p_{ki} - p_{kj} p_{li})(\pi_k \pi'_l - \pi_l \pi'_k). \tag{D.1}$$

Recall our objective of showing that $\pi A^{ij} \pi'^{\top} \geq 0$ for $s_i \geq s_j$ and $\pi A^{ij} \pi'^{\top} = 0$ for $s_i \parallel s_j$. First, consider the case where $s_i \succcurlyeq s_j$. If $s_k \succcurlyeq s_l$, then by $\pi \succcurlyeq_{gr} \pi'$, $\pi_k \pi'_l - \pi_l \pi'_k \geq 0$, and since P is assumed to be GTP$_2$, we have that $p_{lj} p_{ki} - p_{kj} p_{li} \geq 0$, and the corresponding term in the sum is positive (see Eq. (D.1)). Otherwise, if $s_k \parallel s_l$ then $\pi_k \pi'_l - \pi_l \pi'_k = 0$ and the corresponding term in the sum is zero, regardless of the sign of $p_{lj} p_{ki} - p_{kj} p_{li}$. Consequently $\pi A^{ij} \pi'^{\top} \geq 0$ when $s_i \succcurlyeq s_j$. Second, consider the case where $s_i \parallel s_j$. As in the first case, if $s_k \succcurlyeq s_l$ then $\pi_k \pi'_l - \pi_l \pi'_k \geq 0$, but now since $s_i \parallel s_j$, we have that $p_{lj} p_{ki} - p_{kj} p_{li} = 0$ since P is GTP$_2$, resulting in the corresponding term in the sum to be zero. If $s_k \parallel s_l$ then $\pi_k \pi'_l - \pi_l \pi'_k = 0$ and the corresponding term in the sum is zero, regardless of the sign of $p_{lj} p_{ki} - p_{kj} p_{li}$. Consequently $\pi A^{ij} \pi'^{\top} = 0$ when $s_i \parallel s_j$. □

D.3. Proof of Lemma 7.5.1

Proof. For any $\pi \in \Delta(\mathcal{S})$, $u \in \mathcal{U}$, and $y_k, y_l \in \mathcal{Y}$ such that $y_k \succcurlyeq_{\mathcal{Y}} y_l$, $\tau(\pi, u, y_k) \succcurlyeq_{gr} \tau(\pi, u, y_l)$ if and only if (by Definition 7.4.1)

$$\begin{aligned} \tau_i(\pi, u, y_k)\tau_j(\pi, u, y_l) &\geq \tau_j(\pi, u, y_k)\tau_i(\pi, u, y_l) \quad \text{for } s_i \succcurlyeq s_j \\ \tau_i(\pi, u, y_k)\tau_j(\pi, u, y_l) &= \tau_j(\pi, u, y_k)\tau_i(\pi, u, y_l) \quad \text{for } s_i \parallel s_j \end{aligned}$$

for all $y_k \succcurlyeq_{\mathcal{Y}} y_l$. Using the definition of $\tau_i(\pi, u, y)$, Eq. (7.1), we can expand the above expressions to obtain

$$\begin{aligned}
\left(\frac{r_{ik}\sum_{a=1}^{n}\pi_a p_{ai}^u}{\sigma(\pi,u,y_k)}\right)\left(\frac{r_{jl}\sum_{a=1}^{n}\pi_a p_{aj}^u}{\sigma(\pi,u,y_l)}\right) &\geq \left(\frac{r_{jk}\sum_{a=1}^{n}\pi_a p_{aj}^u}{\sigma(\pi,u,y_k)}\right)\left(\frac{r_{il}\sum_{a=1}^{n}\pi_a p_{ai}^u}{\sigma(\pi,u,y_l)}\right) \quad \text{for } s_i \succcurlyeq s_j \\
\left(\frac{r_{iv}\sum_{a=1}^{n}\pi_a p_{ai}^u}{\sigma(\pi,u,y_k)}\right)\left(\frac{r_{jl}\sum_{a=1}^{n}\pi_a p_{aj}^u}{\sigma(\pi,u,y_l)}\right) &= \left(\frac{r_{jk}\sum_{a=1}^{n}\pi_a p_{aj}^u}{\sigma(\pi,u,y_k)}\right)\left(\frac{r_{il}\sum_{a=1}^{n}\pi_a p_{ai}^u}{\sigma(\pi,u,y_l)}\right) \quad \text{for } s_i \parallel s_j
\end{aligned}$$

for all $y_k \succcurlyeq_{\mathcal{Y}} y_l$. Multiplying both sides of the expressions by $\sigma(\pi,u,y_k)\sigma(\pi,u,y_l)$, defined in Eq. (7.2), we obtain

$$\begin{aligned}
\left(r_{ik}\sum_{a=1}^{n}\pi_a p_{ai}^u\right)\left(r_{jl}\sum_{a=1}^{n}\pi_a p_{aj}^u\right) &\geq \left(r_{jk}\sum_{a=1}^{n}\pi_a p_{aj}^u\right)\left(r_{il}\sum_{a=1}^{n}\pi_a p_{ai}^u\right) \quad \text{for } s_i \succcurlyeq s_j \\
\left(r_{ik}\sum_{a=1}^{n}\pi_a p_{ai}^u\right)\left(r_{jl}\sum_{a=1}^{n}\pi_a p_{aj}^u\right) &= \left(r_{jk}\sum_{a=1}^{n}\pi_a p_{aj}^u\right)\left(r_{il}\sum_{a=1}^{n}\pi_a p_{ai}^u\right) \quad \text{for } s_i \parallel s_j
\end{aligned}$$

for all $y_k \succcurlyeq_{\mathcal{Y}} y_l$. Rearranging, the expressions can be equivalently written as

$$\begin{aligned}
(r_{ik}r_{jl} - r_{jk}r_{il})\left(\sum_{a=1}^{n}\pi_a p_{ai}^u\right)\left(\sum_{a=1}^{n}\pi_a p_{aj}^u\right) &\geq 0 \quad \text{for } s_i \succcurlyeq s_j \\
(r_{ik}r_{jl} - r_{jk}r_{il})\left(\sum_{a=1}^{n}\pi_a p_{ai}^u\right)\left(\sum_{a=1}^{n}\pi_a p_{aj}^u\right) &= 0 \quad \text{for } s_i \parallel s_j
\end{aligned}$$

for all $y_k \succcurlyeq_{\mathcal{Y}} y_l$. The above expressions are true if and only if

$$\begin{aligned}
r_{ik}r_{jl} &\geq r_{jk}r_{il} \quad \text{for } s_i \succcurlyeq s_j \\
r_{ik}r_{jl} &= r_{jk}r_{il} \quad \text{for } s_i \parallel s_j
\end{aligned}$$

for all $y_k \succcurlyeq_{\mathcal{Y}} y_l$. By assumption, we have that $r_{ik}r_{jl} = r_{jk}r_{il}$ if either $s_i \parallel s_j$ and $y_k \succcurlyeq_{\mathcal{Y}} y_l$ or $s_i \succcurlyeq s_j$ and $y_k \parallel_{\mathcal{Y}} y_l$, so the above is equivalent to

$$\begin{aligned} r_{ik}r_{jl} &\geq r_{jk}r_{il} \quad \text{for } y_k \succcurlyeq_{\mathcal{Y}} y_l \\ r_{ik}r_{jl} &= r_{jk}r_{il} \quad \text{for } y_k \parallel_{\mathcal{Y}} y_l \end{aligned}$$

for all $s_i \succcurlyeq s_j$, which is $r_i \succcurlyeq_{gr} r_j$ for $s_i \succcurlyeq s_j$. □

D.4. Proof of Lemma 7.5.2

Proof. We need to show that the information state update preserves the generalized MLR order (for a fixed action and observation) if and only if the transition matrix preserves generalized MLR order. For any $u \in \mathcal{U}$, $y_v \in \mathcal{Y}$, and $\pi, \pi' \in \Delta(\mathcal{S})$ such that $\pi \succcurlyeq_{gr} \pi'$, $\tau(\pi, u, y_k) \succcurlyeq_{gr} \tau(\pi', u, y_k)$ if and only if

$$\begin{aligned} \tau_i(\pi, u, y_k)\tau_j(\pi', u, y_k) &\geq \tau_j(\pi, u, y_k)\tau_i(\pi', u, y_k) \quad \text{for all } s_i \succcurlyeq s_j \\ \tau_i(\pi, u, y_k)\tau_j(\pi', u, y_k) &= \tau_j(\pi, u, y_k)\tau_i(\pi', u, y_k) \quad \text{for all } s_i \parallel s_j \end{aligned}$$

for $\pi \succcurlyeq_{gr} \pi'$. The above can be shown to be equivalent to

$$\begin{aligned} r_{ik}r_{jk}\left(\sum_{a=1}^{n} \pi_a p_{ai}^u\right)\left(\sum_{a=1}^{n} \pi'_a p_{aj}^u\right) &\geq r_{jk}r_{ik}\left(\sum_{a=1}^{n} \pi_a p_{aj}^u\right)\left(\sum_{a=1}^{n} \pi'_a p_{ai}^u\right) \quad \text{for } s_i \succcurlyeq s_j \\ r_{ik}r_{jk}\left(\sum_{a=1}^{n} \pi_a p_{ai}^u\right)\left(\sum_{a=1}^{n} \pi'_a p_{aj}^u\right) &= r_{jk}r_{ik}\left(\sum_{a=1}^{n} \pi_a p_{aj}^u\right)\left(\sum_{a=1}^{n} \pi'_a p_{ai}^u\right) \quad \text{for } s_i \parallel s_j \end{aligned}$$

for $\pi \succcurlyeq_{gr} \pi'$. Let $P^u_{\circ,i}$ denote the i'th column of P^u. Dividing both sides of both expressions by $r_{ik}r_{jk}$ (note that $r_{ik}, r_{jk} > 0$ by assumption) yields

$$\begin{aligned} \pi P^u_{\circ,i}\pi' P^u_{\circ,j} &\geq \pi P^u_{\circ,j}\pi' P^u_{\circ,i} \quad \text{for } s_i \succcurlyeq s_j \\ \pi P^u_{\circ,i}\pi' P^u_{\circ,j} &= \pi P^u_{\circ,j}\pi' P^u_{\circ,i} \quad \text{for } s_i \parallel s_j \end{aligned}$$

for $\pi \succcurlyeq_{gr} \pi'$, which is equivalent to $\pi P^u \succcurlyeq_{gr} \pi' P^u$ for $\pi \succcurlyeq_{gr} \pi'$. □

D.5. Proof of Lemma 7.5.3

Proof. Let $P^u_{i,\circ}$ denote the i'th row of matrix P^u. By assumption 1,

$$P^u_{i,\circ} \succcurlyeq_{gs} P^u_{j,\circ}$$

for $s_i \succcurlyeq s_j$. This can be seen by recognizing that, for any $s_i \succcurlyeq s_j$, the degenerate beliefs $v_i, v_j \in \Delta(\mathcal{S})$ (where v_i is a pmf with all mass on element i) satisfy $v_i \succcurlyeq_{gr} v_j$ and noticing that $v_i P^u = P^u_{i,\circ} \succcurlyeq_{gr} P^u_{j,\circ} = v_j P^u$ by Proposition 1 and thus $P^u_{i,\circ} \succcurlyeq_{gs} P^u_{j,\circ}$ by Lemma 7.4.1. By assumption 2, $r_i \succcurlyeq_{gr} r_j$ for all $s_i \succcurlyeq s_j$ and thus $r_i \succcurlyeq_{gs} r_j$ for all $s_i \succcurlyeq s_j$ by Lemma 7.4.1. That is

$$r_i I_J \geq r_j I_J$$

for all $J \in \mathcal{J} = \{J \subseteq \mathcal{Y} \mid y_l \in J, y_k \succcurlyeq_{\mathcal{Y}} y_l \implies y_k \in J\}$. Using the aforementioned Lemma 7.3.1 of stochastic dominance on a partially ordered set, we have that $\sum_{j=1}^{n} p^u_{ij} r_j I_J$ is increasing in i on $(\mathcal{S}, \succcurlyeq)$ for all $J \in \mathcal{J}$. Now, since $\pi \succcurlyeq_{gr} \pi'$ by assumption, $\pi \succcurlyeq_{gs} \pi'$ by

Lemma 7.4.1, and again by Lemma 7.3.1 we have

$$\sum_{i=1}^{n} \pi_i \sum_{j=1}^{n} p_{ij}^{u} r_j I_J \geq \sum_{i=1}^{n} \pi_i' \sum_{j=1}^{n} p_{ij}^{u} r_j I_J$$

for all $J \in \mathcal{J}$. Recall $\sigma(\pi, u, y_k) = \sum_{i=1}^{n} \pi_i \sum_{j=1}^{n} p_{ij}^{u} r_{jk}$, so the above inequality is equivalent to $\sigma(\pi, u) \succeq_{gs} \sigma(\pi', u)$. □

D.6. Proof of Lemma 7.5.4

Proof. The proof proceeds by induction. By assumption $\pi \succeq_{gr} \pi'$ and thus, by Lemma 7.4.1, $\pi \succeq_{gs} \pi'$. Under the assumption that $c(s)$ is increasing in s on $(\mathcal{S}, \succeq)$, Lemma 7.3.1 yields

$$V_T^*(\pi) = \sum_{i=1}^{n} \pi_i c(s_i) \geq \sum_{i=1}^{n} \pi_i' c(s_i) = V_T^*(\pi').$$

Now, assume that $V_{t+1}^*(\pi)$ is increasing on $(\Delta(\mathcal{S}), \succeq_{gr})$, that is, $V_{t+1}^*(\pi) \geq V_{t+1}^*(\pi')$ for $\pi \succeq_{gr} \pi'$ (induction hypothesis). Also, let action u' be optimal in π', that is $u' = g_t^*(\pi')$, so

$$\begin{aligned} V_t^*(\pi') &= \sum_{i=1}^{n} \pi_i' c(s_i, u') + \rho \sum_{k=1}^{m} \sigma(\pi', u', y_k) V_{t+1}^*(\tau(\pi', u', y_k)) \\ &\leq \sum_{i=1}^{n} \pi_i' c(s_i, u) + \rho \sum_{k=1}^{m} \sigma(\pi', u, y_k) V_{t+1}^*(\tau(\pi', u, y_k)) \end{aligned}$$

where $u = g_t^*(\pi)$. By Lemma 7.5.1 and assumptions 4 and 5, $\tau(\pi, u, y)$ is increasing in y on $(\mathcal{Y}, \succeq_{\mathcal{Y}})$ for any $\pi \in \Delta(\mathcal{S})$, $u \in \mathcal{U}$, and by the induction hypothesis, $V_{t+1}^*(\tau(\pi', u, y))$ is also increasing in y on $(\mathcal{Y}, \succeq_{\mathcal{Y}})$. Now by Lemmas 7.3.1 and 7.5.3, we have that

$$\begin{aligned}
&\sum_{i=1}^{n} \pi_i' c(s_i, u) + \rho \sum_{k=1}^{m} \sigma(\pi', u, y_k) V_{t+1}^*(\tau(\pi', u, y_k)) \\
&\leq \sum_{i=1}^{n} \pi_i' c(s_i, u) + \rho \sum_{k=1}^{m} \sigma(\pi, u, y_k) V_{t+1}^*(\tau(\pi', u, y_k)).
\end{aligned} \tag{D.2}$$

Next, note that since $\pi \succeq_{gs} \pi'$ and by assumption 2, $\sum_{i=1}^{n} \pi_i' c(s_i, u) \leq \sum_{i=1}^{n} \pi_i c(s_i, u)$, follows by Lemma 7.3.1. Furthermore, by Lemma 7.5.2 and assumption 3, $\tau(\pi, u, y)$ is increasing in π on $(\Delta(\mathcal{S}), \succeq_{gr})$ for any $u \in \mathcal{U}$, $y_k \in \mathcal{Y}$, and using the induction hypothesis, we have

$$\begin{aligned}
&\sum_{i=1}^{n} \pi_i' c(s_i, u) + \rho \sum_{k=1}^{m} \sigma(\pi, u, y_k) V_{t+1}^*(\tau(\pi', u, y_k)) \\
&\leq \sum_{i=1}^{n} \pi_i c(s_i, u) + \rho \sum_{k=1}^{m} \sigma(\pi, u, y_k) V_{t+1}^*(\tau(\pi, u, y_k)) = V_t^*(\pi).
\end{aligned} \tag{D.3}$$

The result holds by induction. □

References

[Abraham & Efford, 2004] Abraham, S. & Efford, J. (2004). *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*. Technical report, U.S.-Canada Power System Outage Task Force.

[Albright et al., 2010] Albright, D., Brannan, P., & Walrond, C. (2010). *Did Stuxnet Take Out 1,000 Centrifuges at the Natanz Enrichment Plant?* Technical report, Institute for Science and International Security.

[Albright, 1979] Albright, S. C. (1979). Structural results for partially observable Markov decision processes. *Operations Research*, 27(5), 1041–1053.

[Alguacil & Conejo, 2000] Alguacil, N. & Conejo, A. (2000). Multiperiod optimal power flow using Benders decomposition. *IEEE Transactions on Power Systems*, 15(1), 196–201.

[Ammann et al., 2002] Ammann, P., Wijesekera, D., & Kaushik, S. (2002). Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (pp. 217–224).: ACM.

[Aoki & Satoh, 1982] Aoki, K. & Satoh, T. (1982). Economic dispatch with network security constraints using parametric quadratic programming. *IEEE Transactions on Power Apparatus and Systems*, (12), 4548–4556.

[Assante, 2016] Assante, M. (2016). Confirmation of a coordinated attack on the Ukrainian power grid. https://ics.sans.org/blog/2016/01/09/confirmation-of-a-coordinated-attack-on-the-ukrainian-power-grid. 2016-01-09.

[Åström, 1965] Åström, K. J. (1965). Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1), 174.

[Auer et al., 2002] Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3), 235–256.

[Bai et al., 2008] Bai, X., Wei, H., Fujisawa, K., & Wang, Y. (2008). Semidefinite programming for optimal power flow problems. *International Journal of Electrical Power & Energy Systems*, 30(6), 383–392.

[Bakirtzis & Biskas, 2002] Bakirtzis, A. & Biskas, P. (2002). Decentralised DC load flow and applications to transmission management. *IEE Proceedings-Generation, Transmission and Distribution*, 149(5), 600–606.

[Bakirtzis & Biskas, 2003] Bakirtzis, A. G. & Biskas, P. N. (2003). A decentralized solution to the DC-OPF of interconnected power systems. *IEEE Transactions on Power Systems*, 18(3), 1007–1013.

[Baldick et al., 1992] Baldick, R., Kaye, R., & Wu, F. (1992). Electricity tariffs under imperfect knowledge of participant benefits. *IEEE Transactions on Power Systems*, 7(4), 1471–1482.

[Baldick et al., 1999] Baldick, R., Kim, B. H., Chase, C., & Luo, Y. (1999). A fast distributed implementation of optimal power flow. *IEEE Transactions on Power Systems*, 14(3), 858–864.

[Balepin et al., 2003] Balepin, I., Maltsev, S., Rowe, J., & Levitt, K. (2003). Using specification-based intrusion detection for automated response. In *International Workshop on Recent Advances in Intrusion Detection* (pp. 136–154).: Springer.

[Bard, 1988] Bard, J. F. (1988). Short-term scheduling of thermal-electric generators using Lagrangian relaxation. *Operations Research*, 36(5), 756–766.

[Batut & Renaud, 1992] Batut, J. & Renaud, A. (1992). Daily generation scheduling optimization with transmission constraints: A new class of algorithms. *IEEE Transactions on Power Systems*, 7(3), 982–989.

[Bazaraa et al., 2013] Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2013). *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons.

[Bellman, 1955] Bellman, R. (1955). Equipment replacement policy. *Journal of the Society for Industrial and Applied Mathematics*, 3(3), 133–136.

[Bertsekas, 1999] Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena scientific.

[Biskas & Bakirtzis, 2004] Biskas, P. & Bakirtzis, A. (2004). Decentralised security constrained DC-OPF of interconnected power systems. *IEE Proceedings-Generation, Transmission and Distribution*, 151(6), 747–754.

[Biskas et al., 2005] Biskas, P. N., Bakirtzis, A. G., Macheras, N. I., & Pasialis, N. K. (2005). A decentralized implementation of DC optimal power flow on a network of computers. *IEEE Transactions on Power Systems*, 20(1), 25–33.

[Boyd et al., 2011] Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1), 1–122.

[Cain et al., 2012] Cain, M. B., O'Neill, R. P., & Castillo, A. (2012). History of optimal power flow and formulations. Staff paper.

[Carpentier, 1962] Carpentier, J. (1962). Contribution a l'étude du dispatching economique. *Bulletin de la Societe Francaise des Electriciens*, 3(1), 431–447.

[Chao & Peck, 1996] Chao, H. P. & Peck, S. (1996). A market mechanism for electric power transmission. *Journal of Regulatory Economics*, 10, 25–59.

[Chen & Chen, 2003] Chen, S.-D. & Chen, J.-F. (2003). A direct Newton–Raphson economic emission dispatch. *International Journal of Electrical Power & Energy Systems*, 25(5), 411–417.

[Cherepanov & Lipovsky, 2016] Cherepanov, A. & Lipovsky, R. (2016). Blackenergy – what we really know about the notorious cyber attacks. https://www.virusbulletin.com/virusbulletin/2017/07/vb2016-paper-blackenergy-what-we-really-know-about-notorious-cyber-attacks/. 2016-10-07.

[Cherry, 2010] Cherry, S. (2010). How Stuxnet is rewriting the cyberterrorism playbook. https://spectrum.ieee.org/podcast/telecom/security/how-stuxnet-is-rewriting-the-cyberterrorism-playbook. 2010-10-13.

[Christakou et al., 2015] Christakou, K., Tomozei, D.-C., Boudec, J.-Y. L., & Paolone, M. (2015). AC OPF in radial distribution networks-Parts I, II. *arXiv preprint arXiv:1503.06809*.

[Christie & Bose, 1996] Christie, R. D. & Bose, A. (1996). Load frequency control issues in power system operations after deregulation. *IEEE Transactions on Power System*, 11(3), 1191–1200.

[Chung et al., 2011] Chung, K., Kim, B., & Hur, D. (2011). Multi-area generation scheduling algorithm with regionally distributed optimal power flow using alternating direction method. *International Journal of Electrical Power & Energy Systems*, 33(9), 1527–1535.

[Conejo & Aguado, 1998] Conejo, A. & Aguado, J. (1998). Multi-area coordinated decentralized DC optimal power flow. *IEEE Transactions on Power Systems*, 13(4), 1272–1278.

[Conejo et al., 2002] Conejo, A. J., Nogales, F. J., & Prieto, F. J. (2002). A decomposition procedure based on approximate Newton directions. *Mathematical Programming*, 93(3), 495–515.

[Cottle et al., 1992] Cottle, R., Pang, J., & Stone, R. (1992). *The Linear Complementarity Problem*. Classics in Applied Mathematics. SIAM, Philadelphia, PA.

[Dall'Anese et al., 2013] Dall'Anese, E., Zhu, H., & Giannakis, G. B. (2013). Distributed optimal power flow for smart microgrids. *IEEE Transactions on Smart Grid*, 4(3), 1464–1475.

[Department of Homeland Security, 2016] Department of Homeland Security (2016). Industrial control systems cyber emergency response team (ICS-CERT). https://ics-cert.us-cert.gov/. 2016-03-30.

[Derman, 1963] Derman, C. (1963). On optimal replacement rules when changes of state are Markovian. In R. Bellman (Ed.), *Mathematical Optimization Techniques*, volume 396 chapter 9, (pp. 201–210). University of California Press.

[Derman & Sacks, 1960] Derman, C. & Sacks, J. (1960). Replacement of periodically inspected equipment (An optimal optional stopping rule). *Naval Research Logistics (NRL)*, 7(4), 597–607.

[dos Santos & Diniz, 2011] dos Santos, T. N. & Diniz, A. L. (2011). A dynamic piecewise linear model for DC transmission losses in optimal scheduling problems. *IEEE Transactions on Power Systems*, 26(2), 508–519.

[Elgerd, 1973] Elgerd, O. (1973). *Electric Energy Systems Theory: An Introduction.* McGraw-Hill.

[Erseghe, 2014] Erseghe, T. (2014). Distributed optimal power flow using ADMM. *IEEE Transactions on Power Systems*, 29(5), 2370–2380.

[Etherington & Conger, 2016] Etherington, D. & Conger, K. (2016). Large DDoS attacks cause outages at Twitter, Spotify, and other sites. https://techcrunch.com/2016/10/21/many-sites-including-twitter-and-spotify-suffering-outage/. 2016-10-21.

[Falliere et al., 2011] Falliere, N., O'Murchu, L., & Chien, E. (2011). *W32.Stuxnet Dossier.* Technical report, Symantec.

[Fan & Zhang, 1998] Fan, J.-Y. & Zhang, L. (1998). Real-time economic dispatch with line flow and emission constraints using quadratic programming. *IEEE Transactions on Power Systems*, 13(2), 320–325.

[Farivar & Low, 2013a] Farivar, M. & Low, S. H. (2013a). Branch flow model: Relaxations and convexification, Part I. *IEEE Transactions on Power Systems*, 28(3), 2554–2564.

[Farivar & Low, 2013b] Farivar, M. & Low, S. H. (2013b). Branch flow model: Relaxations and convexification, Part II. *IEEE Transactions on Power Systems*, 28(3), 2565–2572.

[Finkle & Skariachan, 2013] Finkle, J. & Skariachan, D. (2013). Target cyber breach hits 40 million payment cards at holiday peak. http://www.reuters.com/article/us-target-breach-idUSBRE9BH1GX20131219. 2013-12-18.

[Foo et al., 2008] Foo, B., Glause, M. W., Howard, G. M., Wu, Y.-S., Bagchi, S., & Spafford, E. H. (2008). Intrusion response systems: A survey. *Information Assurance: Dependability and Security in Networked Systems. Morgan Kaufmann, Burlington.*

[Foo et al., 2005] Foo, B., Wu, Y.-S., Mao, Y.-C., Bagchi, S., & Spafford, E. (2005). ADEPTS: Adaptive intrusion response using attack graphs in an e-commerce environment. In *2005 International Conference on Dependable Systems and Networks* (pp. 508–517).: IEEE.

[Frank et al., 2012a] Frank, S., Steponavice, I., & Rebennack, S. (2012a). Optimal power flow: A bibliographic survey I, formulations and deterministic methods. *Energy Systems*, 3(3), 259–289.

[Frank et al., 2012b] Frank, S., Steponavice, I., & Rebennack, S. (2012b). Optimal power flow: a bibliographic survey II, non-deterministic and hybrid methods. *Energy Systems*, 3(3), 259–289.

[Gabay, 1983] Gabay, D. (1983). Applications of the method of multipliers to variational inequalities. In M. Fortin & R. Glowinski (Eds.), *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*. North-Holland: Amsterdam.

[Gabay & Mercier, 1976] Gabay, D. & Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1), 17–40.

[Galiana et al., 2002] Galiana, F. D., Motto, A. L., Conejo, A. J., & Huneault, M. (2002). Decentralized nodal-price self-dispatch and unit commitment. In B. F. Hobbs, M. H. Rothkopf, R. P. O'Neill, & H. P. Chao (Eds.), *The Next Generation of Electric Power Unit Commitment Models*, volume 36 of *International Series in Operations Research and Management Science* (pp. 271–292). Springer US.

[Girshick & Rubin, 1952] Girshick, M. A. & Rubin, H. (1952). A Bayes approach to a quality control model. *The Annals of mathematical statistics*, (pp. 114–125).

[Glowinski & Marroco, 1975] Glowinski, R. & Marroco, A. (1975). Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2), 41–76.

[Gorenc & Sands, 2017] Gorenc, B. & Sands, F. (2017). *Hacker Machine Interface: The State of SCADA HMI Vulnerabilities*. Technical report, Trend Micro Zero Day Initiative Team.

[Greenberg, 2015] Greenberg, A. (2015). Hackers remotely kill a jeep on the highway – with me in it. https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/. 2015-07-21.

[Horn & Johnson, 1985] Horn, R. & Johnson, C. (1985). *Matrix analysis*. Cambridge University Press.

[Hug-Glanzmann & Andersson, 2009] Hug-Glanzmann, G. & Andersson, G. (2009). Decentralized optimal power flow control for overlapping areas in power systems. *IEEE Transactions on Power Systems*, 24(1), 327–336.

[Iannucci & Abdelwahed, 2016] Iannucci, S. & Abdelwahed, S. (2016). A probabilistic approach to autonomic security management. In *Proceedings of the 13th IEEE International Conference on Autonomic Computing (ICAC)*.

[Iannucci et al., 2016] Iannucci, S., Chen, Q., & Abdelwahed, S. (2016). High-performance intrusion response planning on many-core architectures. In *Computer Communication and Networks (ICCCN), 2016 25th International Conference on* (pp. 1–6).: IEEE.

[Inayat et al., 2016] Inayat, Z., Gani, A., Anuar, N. B., Khan, M. K., & Anwar, S. (2016). Intrusion response systems: Foundations, design, and challenges. *Journal of Network and Computer Applications*, 62, 53–74.

[Jajodia et al., 2005] Jajodia, S., Noel, S., & O?Berry, B. (2005). Topological analysis of network attack vulnerability. In *Managing Cyber Threats* (pp. 247–266). Springer.

[Kamae et al., 1977] Kamae, T., Krengel, U., & O'Brien, G. L. (1977). Stochastic inequalities on partially ordered spaces. *The Annals of Probability*, (pp. 899–912).

[Karlin, 1968] Karlin, S. (1968). *Total positivity*, volume 1. Stanford University Press.

[Karlin & Rinott, 1980] Karlin, S. & Rinott, Y. (1980). Classes of orderings of measures and related correlation inequalities, I. Multivariate totally positive distributions. *Journal of Multivariate Analysis*, 10(4), 467–498.

[Kekatos & Giannakis, 2013] Kekatos, V. & Giannakis, G. B. (2013). Distributed robust power system state estimation. *IEEE Transactions on Power Systems*, 28(2), 1617–1626.

[Khaitan & McCalley, 2015] Khaitan, S. K. & McCalley, J. D. (2015). Design techniques and applications of cyberphysical systems: A survey. *IEEE Systems Journal*, 9(2), 350–365.

[Kheir et al., 2010] Kheir, N., Cuppens-Boulahia, N., Cuppens, F., & Debar, H. (2010). A service dependency model for cost-sensitive intrusion response. In *European Symposium on Research in Computer Security* (pp. 626–642).: Springer.

[Kim & Baldick, 1997] Kim, B. & Baldick, R. (1997). Coarse-grained distributed optimal power flow. *IEEE Transactions on Power Systems*, 12(2), 932–939.

[Kim & Baldick, 2000] Kim, B. & Baldick, R. (2000). A comparison of distributed optimal power flow algorithms. *IEEE Transactions on Power Systems*, 15(2), 599–604.

[Kim et al., 2001] Kim, J., Park, J., Kim, B., Park, J., & Hur, D. (2001). A method of inclusion of security constraints with distributed optimal power flow. *International Journal of Electrical Power & Energy Systems*, 23(3), 189–194.

[Kirchmayer, 1958] Kirchmayer, L. K. (1958). *Economic operation of power systems*, volume 707. Wiley New York.

[Kirschen & Strbac, 2004] Kirschen, D. & Strbac, G. (2004). *Fundamentals of Power System Economics*. John Wiley & Sons.

[Kraning et al., 2013] Kraning, M., Chu, E., Lavaei, J., & Boyd, S. (2013). Dynamic network energy management via proximal message passing. *Foundations and Trends in Optimization*, 1(2), 70–122.

[Kreidl & Frazier, 2004] Kreidl, O. P. & Frazier, T. M. (2004). Feedback control applied to survivability: A host-based autonomic defense system. *IEEE Transactions on Reliability*, 53(1), 148–166.

[Kumar & Varaiya, 1986] Kumar, P. R. & Varaiya, P. (1986). *Stochastic systems: Estimation, identification, and adaptive control*. Prentice Hall Englewood Cliffs, NJ.

[Kurniawati et al., 2008] Kurniawati, H., Hsu, D., & Lee, W. S. (2008). SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*: Zurich, Switzerland.

[Langner, 2013] Langner, R. (2013). *To kill a centrifuge: A technical analysis of what Stuxnet's creators tried to achieve*. Technical report, The Langner Group.

[Lavaei & Low, 2012] Lavaei, J. & Low, S. (2012). Zero duality gap in optimal power flow problem. *IEEE Transactions on Power Systems*, 27(1), 92 –107.

[Lavaei & Sojoudi, 2012] Lavaei, J. & Sojoudi, S. (2012). Competitive equilibria in electricity markets with nonlinearities. In *American Control Conference (ACC)* (pp. 3081–3088).

[Lee et al., 2002] Lee, W., Fan, W., Miller, M., Stolfo, S. J., & Zadok, E. (2002). Toward cost-sensitive modeling for intrusion detection and response. *Journal of Computer Security*, 10(1-2), 5–22.

[Lewandowski et al., 2001] Lewandowski, S. M., Van Hook, D. J., O'Leary, G. C., Haines, J. W., & Rossey, L. M. (2001). SARA: Survivable autonomic response architecture. In *DARPA Information Survivability Conference & Exposition II*, volume 1 (pp. 77–88).: IEEE.

[Liu & Man, 2005] Liu, Y. & Man, H. (2005). Network vulnerability assessment using Bayesian networks. In *Defense and Security* (pp. 61–71).: International Society for Optics and Photonics.

[Lovejoy, 1987] Lovejoy, W. S. (1987). Some monotonicity results for partially observed Markov decision processes. *Operations Research*, 35(5), 736–743.

[Low, 2013] Low, S. H. (2013). Convex relaxation of optimal power flow: a tutorial. In *Bulk Power System Dynamics and Control-IX Optimization, Security and Control of the Emerging Power Grid (IREP), 2013 IREP Symposium* (pp. 1–15).: IEEE.

[Magnusson et al., 2014] Magnusson, S., Weeraddana, P., & Fischione, C. (2014). A distributed approach for the optimal power flow problem based on ADMM and sequential convex approximations. *arXiv preprint arXiv:1401.4621*.

[Mas-Colell et al., 1995] Mas-Colell, A., Whinston, M., & Green, J. (1995). *Microeconomic Theory*. Oxford University Press.

[Miehling et al., 2015] Miehling, E., Rasouli, M., & Teneketzis, D. (2015). Optimal defense policies for partially observable spreading processes on Bayesian attack graphs. In *Proceedings of the Second ACM Workshop on Moving Target Defense* (pp. 67–76).: ACM.

[Mirsky, 2012] Mirsky, L. (2012). *An Introduction to Linear Algebra*. Courier Dover Publications.

[Molzahn et al., 2013] Molzahn, D. K., Holzer, J. T., Lesieutre, B. C., & DeMarco, C. L. (2013). Implementation of a large-scale optimal power flow solver based on semidefinite programming. *IEEE Transactions on Power Systems*, 28(4), 3987–3998.

[Morgan, 2015] Morgan, P. B. (2015). *An Explanation of Constrained Optimization for Economists*. University of Toronto Press.

[Mosca, 2013] Mosca, U. (2013). A novel distributed approach for optimal power flow problem in smart grids. Master's thesis, KTH Royal Institute of Technology.

[Motto et al., 2002a] Motto, A., Galiana, F., Conejo, A., & Huneault, M. (2002a). On Walrasian equilibrium for pool-based electricity markets. *IEEE Transactions on Power Systems*, 17(3), 774–781.

[Motto et al., 2002b] Motto, A. L., Galiana, F. D., Conejo, A. J., & Arroyo, J. M. (2002b). Network-constrained multiperiod auction for a pool-based electricity market. *IEEE Transactions on Power Systems*, 17(3), 646–653.

[Nguyen et al., 2012] Nguyen, D. T., Negnevitsky, M., & De Groot, M. (2012). Walrasian market clearing for demand response exchange. *IEEE Transactions on Power System*, 27(1), 535–544.

[Noel & Jajodia, 2004] Noel, S. & Jajodia, S. (2004). Managing attack graph complexity through visual hierarchical aggregation. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security* (pp. 109–118).: ACM.

[Nogales et al., 2003] Nogales, F. J., Prieto, F. J., & Conejo, A. J. (2003). A decomposition methodology applied to the multi-area optimal power flow problem. *Annals of Operations Research*, 120(1-4), 99–116.

[Ongsakul & Petcharaks, 2004] Ongsakul, W. & Petcharaks, N. (2004). Unit commitment by enhanced adaptive Lagrangian relaxation. *IEEE Transactions on Power System*, 19(1), 620–628.

[Pandya & Joshi, 2008] Pandya, K. & Joshi, S. (2008). A survey of optimal power flow methods. *Journal of Theoretical & Applied Information Technology*, 4(5).

[Papadaskalopoulos & Strbac, 2013] Papadaskalopoulos, D. & Strbac, G. (2013). Decentralized participation of flexible demand in electricity markets - Part I: Market mechanism. *IEEE Transactions on Power System*, 28(4), 3658–3666.

[Peng & Low, 2014] Peng, Q. & Low, S. H. (2014). Distributed algorithm for optimal power flow on a radial network. *arXiv preprint arXiv:1404.0700*.

[Porras & Neumann, 1997] Porras, P. A. & Neumann, P. G. (1997). EMERALD: Event monitoring enabling response to anomalous live disturbances. In *Proceedings of the 20th National Information Systems Security Conference* (pp. 353–365).

[Porteus, 1975] Porteus, E. L. (1975). On the optimality of structured policies in countable stage decision processes. *Management Science*, 22(2), 148–157.

[Poulsen, 2004] Poulsen, K. (2004). Software bug contributed to blackout. http://www.securityfocus.com/news/8016. 2004-02-11.

[Pultarova, 2016] Pultarova, T. (2016). News briefing: Cyber security-Ukraine grid hack is wake-up call for network operators. *Engineering & Technology*, 11(1), 12–13.

[Ragsdale et al., 2000] Ragsdale, D. J., Carver, C., Humphries, J. W., & Pooch, U. W. (2000). Adaptation techniques for intrusion detection and intrusion response systems. In *2000 IEEE International Conference on Systems, Man, and Cybernetics*, volume 4 (pp. 2344–2349).: IEEE.

[Rajkumar et al., 2010] Rajkumar, R. R., Lee, I., Sha, L., & Stankovic, J. (2010). Cyber-physical systems: the next computing revolution. In *Proceedings of the 47th Design Automation Conference* (pp. 731–736).: ACM.

[Rockafellar, 1970] Rockafellar, R. (1970). *Convex Analysis*. Princeton University Press.

[Rosenfield, 1976a] Rosenfield, D. (1976a). Markovian deterioration with uncertain information. *Operations Research*, 24(1), 141–155.

[Rosenfield, 1976b] Rosenfield, D. (1976b). Markovian deterioration with uncertain information – a more general model. *Naval Research Logistics (NRL)*, 23(3), 389–405.

[Ross et al., 2008] Ross, S., Pineau, J., Paquet, S., & Chaib-Draa, B. (2008). Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, 32, 663–704.

[Ross, 1971] Ross, S. M. (1971). Quality control under markovian deterioration. *Management Science*, 17(9), 587–596.

[Ryutov et al., 2003] Ryutov, T., Neuman, C., Dongho, K., & Li, Z. (2003). Integrated access control and intrusion detection for web servers. *IEEE Transactions on Parallel and Distributed Systems*, 14(9), 841–850.

[Schneier, 1999] Schneier, B. (1999). Attack trees. *Dr. Dobb's Journal*, 24(12), 21–29.

[Security Response Team, 2017] Security Response Team (2017). What you need to know about the WannaCry ransomware. https://www.symantec.com/blogs/threat-intelligence/wannacry-ransomware-attack. 2017-10-23.

[Shameli-Sendi et al., 2012] Shameli-Sendi, A., Ezzati-Jivan, N., Jabbarifar, M., & Dagenais, M. (2012). Intrusion response systems: survey and taxonomy. *International Journal of Computer Science and Network Security*, 12(1), 1–14.

[Sheyner et al., 2002] Sheyner, O., Haines, J., Jha, S., Lippmann, R., & Wing, J. M. (2002). Automated generation and analysis of attack graphs. In *Proceedings of IEEE Symposium on Security and Privacy* (pp. 273–284).

[Silver & Veness, 2010] Silver, D. & Veness, J. (2010). Monte-Carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems* (pp. 2164–2172).

[Stoft, 2002] Stoft, S. (2002). *Power System Economics: Designing Markets for Electricity.* IEEE Press. Wiley.

[Stott et al., 1987] Stott, B., Alsac, O., & Monticelli, A. (1987). Security analysis and optimization. *Proceedings of the IEEE*, 75(12), 1623–1644.

[Stott et al., 2009] Stott, B., Jardim, J., & Alsaç, O. (2009). DC power flow revisited. *IEEE Transactions on Power Systems*, 24(3), 1290–1300.

[Stoyan & Daley, 1983] Stoyan, D. & Daley, D. J. (1983). Comparison methods for queues and other stochastic models. *John Wiley & Sons, New York, NY.*

[Šulc et al., 2014] Šulc, P., Backhaus, S., & Chertkov, M. (2014). Optimal distributed control of reactive power via the alternating direction method of multipliers. *IEEE Transactions on Energy Conversion*, 29(4), 968–977.

[Sun et al., 2013] Sun, A., Phan, D., & Ghosh, S. (2013). Fully decentralized AC optimal power flow algorithms. In *2013 IEEE Power and Energy Society General Meeting (PES)* (pp. 1–5).

[Topkis, 1978] Topkis, D. M. (1978). Minimizing a submodular function on a lattice. *Operations Research*, 26(2), 305–321.

[Toth & Kruegel, 2002] Toth, T. & Kruegel, C. (2002). Evaluating the impact of automated intrusion response mechanisms. In *18th Annual Computer Security Applications Conference, 2002. Proceedings.* (pp. 301–310).: IEEE.

[Trias, 2012] Trias, A. (2012). The holomorphic embedding load flow method. In *2012 IEEE Power and Energy Society General Meeting* (pp. 1–8).: IEEE.

[White, 1979] White, C. C. (1979). Optimal control-limit strategies for a partially observed replacement problem. *International Journal of Systems Science*, 10(3), 321–332.

[White, 1980] White, C. C. (1980). Monotone control laws for noisy, countable-state Markov chains. *European Journal of Operational Research*, 5(2), 124–132.

[White et al., 1996] White, G. B., Fisch, E. A., & Pooch, U. W. (1996). Cooperating security managers: A peer-based intrusion detection system. *IEEE Network*, 10(1), 20–23.

[Whitt, 1979] Whitt, W. (1979). A note on the influence of the sample on the posterior distribution. *Journal of the American Statistical Association*, 74(366a), 424–426.

[Wood & Wollenberg, 2012] Wood, A. J. & Wollenberg, B. F. (2012). *Power Generation, Operation, and Control*. John Wiley & Sons.

[Wu et al., 1996] Wu, F., Varaiya, P., Spiller, P., & Oren, S. (1996). Folk theorems on transmission access: Proofs and counterexamples. *Journal of Regulatory Economics*, 10(1), 5–23.

[Wu & Varaiya, 1999] Wu, F. F. & Varaiya, P. (1999). Coordinated multilateral trades for electric power networks: theory and implementation. *International Journal of Electrical Power & Energy Systems*, 21(2), 75–102.

[Yamanouchi, 2016] Yamanouchi, K. (2016). Delta computer risk 'went undetected,' CEO says. http://www.myajc.com/business/delta-computer-risk-went-undetected-ceo-says/V2H5CYl4Nsbn57x8T5ISvM/. 2016-08-10.

[Zetter, 2016] Zetter, K. (2016). Inside the cunning, unprecedented hack of Ukraine's power grid. https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/. 2016-03-03.

[Zhuang & Galiana, 1988] Zhuang, F. & Galiana, F. D. (1988). Towards a more rigorous and practical unit commitment by Lagrangian relaxation. *IEEE Transactions on Power System*, 3(2), 763–773.

[Zimmerman et al., 2011] Zimmerman, R. D., Murillo-Sánchez, C. E., & Thomas, R. J. (2011). Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1), 12–19.

[Zonouz et al., 2014] Zonouz, S. A., Khurana, H., Sanders, W. H., & Yardley, T. M. (2014). RRE: A game-theoretic intrusion response and recovery engine. *IEEE Transactions on Parallel and Distributed Systems*, 25(2), 395–406.