Computational Models of Competitive Business Development under Imperfect Information

A Dissertation Presented
by
Benjamin P. Harris
to
The Department of Mechanical and Industrial Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Industrial Engineering

Northeastern University
Boston, Massachusetts

March 2017
This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations
To my family, especially my beautiful wife Charlotte and sons Elliott and Gideon.
Contents

List of Figures vi

List of Tables vii

Acknowledgments viii

Abstract ix

1 Introduction 1

1.1 Introduction and Terminology ........................................ 1
1.2 Research Motivation .................................................... 3
1.3 Purpose and Scope ....................................................... 3
1.4 Overview of Results ..................................................... 5
    1.4.1 Large Scale Business ............................................. 5
    1.4.2 Markov Property Shortcomings .................................. 5
    1.4.3 Network Formulation of Games .................................. 5
    1.4.4 Markov Chain Deflection ....................................... 6

2 Literature Review 7

2.1 Game Theory ............................................................ 7
    2.1.1 Probabilistic Strategy Profiles ................................ 11
    2.1.2 Markov Strategies ............................................... 12
    2.1.3 Equilibrium Solutions .......................................... 12
2.2 Operations Research ................................................... 13
    2.2.1 Linear Programming ............................................. 13
    2.2.2 Markov Chains ................................................... 14
2.3 Graph Theory ........................................................... 15
    2.3.1 History and Terminology ....................................... 16
    2.3.2 Computability .................................................... 16
    2.3.3 Deterministic and Stochastic Formulation .................... 17
2.4 Mechanism Design ...................................................... 17
2.5 Portfolio Optimization .................................................. 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.1 Terminology and Computation</td>
<td>20</td>
</tr>
<tr>
<td>2.5.2 Value at Risk</td>
<td>23</td>
</tr>
<tr>
<td>2.6 Product Platforms</td>
<td>24</td>
</tr>
<tr>
<td>2.7 Customer Relationship Management</td>
<td>28</td>
</tr>
<tr>
<td>2.8 Summary of Literature Review</td>
<td>29</td>
</tr>
<tr>
<td>3 Winning Business on the Large Scale</td>
<td>30</td>
</tr>
<tr>
<td>3.1 Firm Decisions</td>
<td>30</td>
</tr>
<tr>
<td>3.2 Definition of Strategies</td>
<td>33</td>
</tr>
<tr>
<td>3.3 Technology Portfolio Network</td>
<td>34</td>
</tr>
<tr>
<td>3.4 Optimization of Investment</td>
<td>34</td>
</tr>
<tr>
<td>3.4.1 Capability Optimization</td>
<td>35</td>
</tr>
<tr>
<td>3.4.2 Investment Optimization</td>
<td>37</td>
</tr>
<tr>
<td>3.5 Corporate and Competitive Strategy Computation</td>
<td>39</td>
</tr>
<tr>
<td>3.5.1 Market Spaces</td>
<td>39</td>
</tr>
<tr>
<td>3.5.2 Relationship to Product Platforms</td>
<td>40</td>
</tr>
<tr>
<td>3.5.3 Production Strategy Networks</td>
<td>41</td>
</tr>
<tr>
<td>3.5.4 Small Example</td>
<td>41</td>
</tr>
<tr>
<td>3.6 External or Environmental Factors</td>
<td>42</td>
</tr>
<tr>
<td>3.7 Closing Remarks</td>
<td>43</td>
</tr>
<tr>
<td>4 A Higher-Order Markov Property</td>
<td>45</td>
</tr>
<tr>
<td>4.1 Introduction to Markov Processes</td>
<td>45</td>
</tr>
<tr>
<td>4.2 Importance in the Development of Stochastic Processes</td>
<td>46</td>
</tr>
<tr>
<td>4.3 Limitations in Stochastic Model Usage</td>
<td>47</td>
</tr>
<tr>
<td>4.4 Results from Computational Linguistics</td>
<td>48</td>
</tr>
<tr>
<td>4.5 Notional Computational Example</td>
<td>51</td>
</tr>
<tr>
<td>4.6 Data-Centric Computational Example</td>
<td>52</td>
</tr>
<tr>
<td>4.7 Data Sizing to Enable Effective Computation</td>
<td>56</td>
</tr>
<tr>
<td>4.8 Closing Remarks</td>
<td>60</td>
</tr>
<tr>
<td>5 Markov Chain Deflection</td>
<td>61</td>
</tr>
<tr>
<td>5.1 Product Network Interdiction Concepts</td>
<td>61</td>
</tr>
<tr>
<td>5.2 Literature Review</td>
<td>62</td>
</tr>
<tr>
<td>5.3 Mathematical Model</td>
<td>63</td>
</tr>
<tr>
<td>5.4 Market Growth</td>
<td>64</td>
</tr>
<tr>
<td>5.4.1 Market Share Impacts</td>
<td>64</td>
</tr>
<tr>
<td>5.4.2 Linear Programming Solution</td>
<td>69</td>
</tr>
<tr>
<td>5.4.3 Long-Run Dynamics</td>
<td>71</td>
</tr>
<tr>
<td>5.4.4 Computational Example</td>
<td>72</td>
</tr>
<tr>
<td>5.4.5 Stiffness Vector Computation</td>
<td>73</td>
</tr>
<tr>
<td>5.5 Discussion</td>
<td>75</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The Markowitz Bullet</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Corporate and Competitive Strategy Interaction</td>
<td>31</td>
</tr>
<tr>
<td>3.2</td>
<td>Stochastic Request Flow</td>
<td>32</td>
</tr>
<tr>
<td>3.3</td>
<td>Market Space Interaction</td>
<td>42</td>
</tr>
<tr>
<td>3.4</td>
<td>Shared Product Feature Network</td>
<td>43</td>
</tr>
<tr>
<td>3.5</td>
<td>Market Space Separation</td>
<td>44</td>
</tr>
<tr>
<td>3.6</td>
<td>Market Space Integration</td>
<td>44</td>
</tr>
<tr>
<td>4.1</td>
<td>Data Size vs. n-gram Length</td>
<td>59</td>
</tr>
<tr>
<td>4.2</td>
<td>Logarithmic Relationship of Data Sizing</td>
<td>59</td>
</tr>
<tr>
<td>5.1</td>
<td>Deflection Feasible Solutions in 2 Dimensions</td>
<td>68</td>
</tr>
<tr>
<td>5.2</td>
<td>Stock Pricing Example</td>
<td>74</td>
</tr>
<tr>
<td>6.1</td>
<td>Two possible “sub-region” solutions</td>
<td>79</td>
</tr>
<tr>
<td>6.2</td>
<td>Undirected Graph with Subregions</td>
<td>80</td>
</tr>
<tr>
<td>6.3</td>
<td>Equivalent undirected graph with reduced payoffs</td>
<td>81</td>
</tr>
<tr>
<td>6.4</td>
<td>Sub-region graph for Keolis vs. MBCR</td>
<td>93</td>
</tr>
<tr>
<td>6.5</td>
<td>Computational Arc Frequency</td>
<td>94</td>
</tr>
<tr>
<td>6.6</td>
<td>Solution frequency using Monte Carlo simulation</td>
<td>95</td>
</tr>
<tr>
<td>6.7</td>
<td>Iteration time for algorithm implementation</td>
<td>96</td>
</tr>
<tr>
<td>7.1</td>
<td>Graphical Integration of Computational Models</td>
<td>102</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Chamberlain’s 7 Propositions on Strategy ........................................ 11
2.2 Technology Portfolio Components .................................................. 22
6.1 Perturbation Value Computations .................................................... 88
6.2 Keolis vs. MBCR Operating Criteria ............................................... 92
6.3 High frequency arcs and relative appearance values ...................... 95
Acknowledgments

The completion of this work could not have happened without the support and guidance of a number of people. I must acknowledge the invaluable support and advice from Prof. Sagar Kamarthi, who has advised me through this process. His advice in matters both academic and career-oriented has been extremely illuminating.

I also want to acknowledge the support and advice of Dr. Neal Mackertich. Neal’s encouragement and support through this program have helped to shape both my academic and professional goals. Neal’s willingness to be a sounding board for ideas, however outlandish, was formative in my ability to envision and solve problems.

I would like to acknowledge the help of Prof. Emanuel Melachrinoudis, both in being willing to serve on my committee, but also for giving me the teaching opportunities that I have been afforded during my time at Northeastern University. Teaching runs in my family, and I have Manny to thank for allowing me to explore why that is.

Lastly, I want to acknowledge the support of Srinivasan Radhakrishnan. Sri’s helpful conversations shaped my ideas, and his strength in areas not my own were helpful and insightful.
Abstract

Computational Models of Competitive Business Development under Imperfect Information

by

Benjamin P. Harris

Doctor of Philosophy in Industrial Engineering

Northeastern University, March 2017

Dr. Sagar Kamarthi, Adviser

Governments and other large organizations are increasingly finding themselves contracting products and services to firms whose specialties align more closely than the original firm. To accomplish this, a number of resources are available for firms seeking to do business with large firms or governmental organizations. These customers solicit bids from prospective suppliers to meet their product or service needs, but often pay little attention to the interactions amongst the competitors during the competition. This can cause the products and services provided to be sub-optimal either in performance or cost. From the perspective of the competitor, I present a number of models by which a complex competition can be envisioned which lead to improved decision-making even when a competitor does not possess complete information.

Chapter 1 provides an introduction to the problem of business decision making under incomplete information. We present a number of real-world data points to reinforce the motivation for this research.

Chapter 2 provides a literature review of the qualitative and quantitative concepts used in this dissertation. The literature review is of course not meant to be exhaustive, and the interested reader may refer to the cited texts for additional information.

Chapter 3 details the concept of a technology portfolio, and how firms make macro-level decisions. The addition or removal of a technology from a firm affects its ability to compete across a spectrum of products. I also investigate investment in existing capabilities and how
that process can improve firm competitiveness. A number of computational examples are presented to demonstrate practical application.

Chapter 4 investigates some shortcomings associated with using the Markovian property in stochastic models for decision making. While usually assumed for computational simplicity, I show a number of instances where the inclusion of the Markovian property causes a sub-optimal set of decisions. I also provide some rules-of-thumb to detect where the Markov property should be used. This chapter concludes with some computational results demonstrating where assuming the Markov property results in sub-optimal decisions, and how the probabilistic definition can be adjusted to provide additional accuracy.

Chapter 5 presents a new method for analyzing interdictions in stochastic processes. Borrowing concepts from Mechanical Engineering, this chapter show how discrete-time Markov Chains can be analyzed much a cantilever beam. Doing so provides an analyst with another tool for estimating the impact of interdictions like research and development or advertising. Several computational examples are also presented.

Chapter 6 shows a conceptual framework for analyzing complex competitions. I proposed constructing an undirected graph to model the relationships between decisions in a competitive landscape. In doing so, graph theory and network science methods can be brought to bear and create a holistic view of the competition in order to find optimal decisions.

Chapter 7 provides a summary of the above results as well as some concluding remarks. Areas for future research are also included.
Chapter 1

Introduction

“There will be very few occasions when you are absolutely certain about something. You will consistently be called upon to make decisions with limited information. That being the case, your goal should not be to eliminate uncertainty. Instead, you must develop the art of being clear in the face of uncertainty.” - Andy Stanley

1.1 Introduction and Terminology

Uncertainty surrounds humanity in nearly all of the decisions that we make. Acknowledgment of this and the formulation of approaches to accommodate intrinsic uncertainty rely on a notion of rationality on the part of the decision-maker. Fudenberg and Tirole [31, pp. 48-49] detail the game-theoretic version of rationality as follows:

"The starting point of iterated strict dominance is the observation that a rational player will never play a strictly dominated strategy. The starting point of rationalizability is the complementary question: What are all the strategies that a rational player could play? The answer is that a rational player will use only those strategies that are best responses to some beliefs he might have about the strategies of his opponents. Or, to use the contrapositive, a player cannot reasonably play a strategy that is not a best response to some beliefs about his opponents’ strategies."

While this definition has enabled much of modern game-theoretic research, I propose additional considerations based on my industry observations and experience. There are compe-
CHAPTER 1. INTRODUCTION

tations and games that do not occur in a static environment. The requirements and preferences of customers change over time, so the evaluation of any given strategy should change as well. If a rational competitor finds that he/she does possess a dominant strategy, they are left with several choices:

1. Do not participate in the game. If every instance of the game result in a player losing, it does not follow rationally to continue playing unless compelled to do so.

2. Change the circumstances of the game. This is where I focus in this dissertation with what I refer to as interventions.

The process of decision-making falls under two major definitions of scope. First, humans make decisions under assumption that what is currently known is all that can be known and is sufficient for the decision at hand. We refer to this as deterministic decision making. Second, humans make decisions based on the information available to them at the time the decision must be made, acknowledging that additional information may later become available. We accordingly refer to this as stochastic decision making. When considering decision making in competitive business environments, both the deterministic and stochastic methods have been applied in the past, and the strengths and weaknesses of each bear elaboration.

1. Deterministic models, by definition, assuming that all model inputs are known with certainty. This is of course an assumption, one that does not often hold perfectly in practice.

2. Stochastic models, on the other hand, acknowledge uncertainty in a number of inputs and/or processes. This creates a potentially more realistic model at the expense of solution precision.

A number of differences separate theoretical and practical models:

- Stochastic and game-theoretic models are brought to bear on these decision processes.

- If, in a game of $n$ different competitors, one competitor finds themselves in a losing position, they will rationally attempt to change the circumstances of the game.
CHAPTER 1. INTRODUCTION

- This can take a number of forms (not exhaustive):
  - Modifying the priorities of the customers in the game.
  - Changing the valuation of the strategies being considered.
  - Affecting environmental changes in order to make a competitor’s solution less attractive to a customer.

Interventions are a major component of this research, and I seek to investigate equilibrium and optimal points when intervention is allowed. Understanding this phenomenon is critical to enable effective decision making when imperfect information and rational competitors are present in the game. A number of terms will be used consistently throughout this dissertation, and precise definitions will add clarity. Competitions between \( n \) players are examined, with each of the \( n \) competitors seeking to be chosen to provide a product or service from a customer. In the Game Theory literature, the customer is referred to as the designer.

1.2 Research Motivation

This research is being conducted with the support and under the auspices of the Raytheon Integrated Defense Systems Advanced Study Program, hereafter referenced as the IDS ASP. Understanding customer motivation is a key component to any marketing and business strategy. As customers in every industry have learned, affecting product change is initially expensive, and becomes exponentially more so as a product grows in maturity.

1.3 Purpose and Scope

The main purpose of this dissertation and the research contained therein is to provide a mathematical method for organizations to utilize collected competitive intelligence to illuminate strategic impacts on a corporate scale. Three major purposes in this research program are:
CHAPTER 1. INTRODUCTION

- Collect and take action in response to competitive and industrial intelligence data gained through predictive mathematical tools or open-source information.

- Mathematically integrate corporate competitive strategy in a given firm with optimal project-level resource allocation.

- Provide feedback mechanisms for the above two items so that a firm can assess how efficient these efforts are.
  
  – This research program lays the mathematical foundation for machine learning algorithms to provide feedback on the methods in the future.

The scope of this research will include the numerical framework for collecting and synthesizing competitive and industrial intelligence data, the mathematical foundations of integrating corporate strategy with project execution, and feedback mechanisms with which to discern the effectiveness of the algorithms presented. Items beyond the scope of this research are specific industrial implementation plans, which would likely contain proprietary data. The relationships between major competitors in all industries are complex, as firms participate on programs in both prime contractor and subcontractor roles, often times completing work for the benefit of a competitor. The rationale behind this allowance is simple: a contractor would rather maintain some program involvement and accrue some profit than none at all! This behavior also provides the subcontractors with program and system knowledge in the event that contracts re-emerge to be completed on again. Over the course of long contracts (lifetimes of > 25 years), the support and maintenance responsibilities may be negotiated and re-competed a number of times. Large firms utilize acquisition and divestiture as a competitive methodology. In 1999 the Nissan and Renault companies formed a partnership with the intent of improving competitiveness in individual market spaces as well as sharing resources in combined markets. Similar reasons underlie the 1998 merger of Daimler-Benz and Chrysler, both to give Daimler-Benz direct US market access but to additionally open Chrysler to overseas markets. We observe a similar pattern of mergers and acquisitions in many other industries as well, which lends credence to this research program. Applying computational methodologies to these types of decisions will pay long-term dividends to firms.
CHAPTER 1. INTRODUCTION

1.4 Overview of Results

This section provides a brief overview of the results presented in the remainder of this dissertation. A summary of the results is presented here, with theoretical and computational details provided in the chapters and appendices.

1.4.1 Large Scale Business

Porter’s definition of strategy [75] delineates between the business unit allocation choices made by leadership in a firm (called corporate strategy) and the individual competitive decisions made by those business units (called competitive strategy). In this chapter I extend the concept of product platforms to incorporate both the corporate and competitive facets of strategic management and development. The chapter also discusses the nature of internal and external factors and incentives and how they impact optimal strategic decision making. Finally, I presented an optimization model for planning both corporate and competitive strategy when the information received from a market space is imperfect.

1.4.2 Markov Property Shortcomings

The Markov property is utilized in a broad cross-section of decision models to simplify the inclusion of information and increase the computational tractability of the model. However, most mathematical descriptions of the Markov Property reduce the state definition to a single value and in the process remove information that may be pertinent to the decision process. This chapter investigates the impact of extending the markov state definition and demonstrates several scenarios where the scalar version of the Markov Property is inadequate and may lead to erroneous conclusions.

1.4.3 Network Formulation of Games

This chapter presents corporate and competitive strategy problems in the context of undirected graphs. This mathematical formulation enables the computational analysis of the problems using the gamut of graph theory tools. I present a basic overview of the
CHAPTER 1. INTRODUCTION

novel additions, sensitivity analysis to accompany the graph formulation, and a number of computational examples showing how the formulation can be applied to real-world strategy problems.

1.4.4 Markov Chain Deflection

Following the theme of interventions in stochastic models from the previous chapters, this chapter proposes a method for measuring the impact of strategic investment or environment manipulation. Given that each row of a discrete-time Markov Chain represents a probability distribution from one state $i$ to another state $j$, I demonstrate a method for shifting the probabilities in the distribution to reflect an intervention in the system. I also provide a linear programming formulation for the method, which identifies optimal investment values and limitations.
Chapter 2

Literature Review

As indicated by the title of this dissertation, a number of fields have proven helpful as I progress in answering business questions under incomplete information. This literature review will provide an overview of the pertinent topics and their mathematical foundations. This review is not exhaustive, and the curious reader is referred to the provided references for additional information.

2.1 Game Theory

The theory of games saw its formal introduction as a field with John von Neumann’s 1928 paper [67], where he used Brouwer’s Fixed-point theorem to estimate equilibrium points on compact, convex sets. Von Neumann’s work in Game Theory reached its most notable accomplishment with the 1944 publishing of Theory of Games and Economic Behavior [99], with Oskar Morgenstern. In the 1950’s, mathematicians began to discuss the foundations of problems such as the prisoner’s dilemma, and companies like the RAND corporation began to take an interest into how the field could be applied to problems of national defense. The following survey provides an outline of game theoretic terms and concepts used in this dissertation. The definitions are primarily taken from the Fudenberg and Tirole textbook [31].

1. Game Types
CHAPTER 2. LITERATURE REVIEW

(a) Games of complete information - also referred to as common knowledge, the concept refers to the scenario when competitors in a game know the structure of the strategic form, the strategies of the opponent, that their opponents knows that they know, and so on ad infinitum.

(b) Games of incomplete information - referring to the scenario when some players do not know the payoffs or strategic capabilities of the others. The presence of complete information is sometimes used as an approximation to games of incomplete information.

(c) Games of imperfect information - imperfect information refers to circumstances when the global objectives of the game are known, but the specific capabilities of each player may not be known. The theoretical connections between games of incomplete and imperfect information were laid out by Harsanyi [40–42] in what would eventually lead to the theory of Bayesian games.

2. Payoff Types

(a) Zero-sum - when the sum of the utilities for any given strategy combination sums to 0, the game is said to be zero-sum. Intuitively, for each instantiation of the game, what one players wins the other loses.

(b) Non Zero-sum - when the sum of the utilities for a given strategy combination does not sum to 0, the game is non-zero-sum. Many of the games considered in this dissertation fall under this category.

3. Game Representations

(a) Normal Form Representation - also called the strategic form, this game representation has three components: the set of players $i \in I$, the pure-strategy space $S_i$ for each player $i$, and payoff functions $u_i$ that assigns each $i$ a utility corresponding to each strategy $s \in S$ [31]. A basic normal-form game (non-
CHAPTER 2. LITERATURE REVIEW

zero-sum) is shown below:

\[
\begin{bmatrix}
  s_1 & s_2 & \cdots & s_m \\
  s_1 & (u_1, u_1) & (u_1, u_2) & \cdots & (u_1, u_m) \\
  s_2 & (u_2, u_1) & (u_2, u_2) & \cdots & (u_2, u_m) \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  s_n & (u_n, u_1) & (u_n, u_2) & \cdots & (u_n, u_m)
\end{bmatrix}
\] (2.1)

The strategies of player 1 (n of them) are shown down the column to the left, while the strategies of player 2 (m of them) are shown in the row across the top of the matrix.

(b) Extensive Form Representation - this game representation changes the implicit assumption of normal-form games that each competitor chooses a strategy simultaneously. The extensive form of the game represents the plays of the players in sequential form, so one player can observe the choice of another before making their own choice. The most common visual representation of this game form is a decision tree.

(c) Network Representation - This representation applies to complex scenarios when a number of interrelated decisions must be made in order to form a strategy. In this dissertation I present a number of mathematical methods for analyzing games represented in this format. A game represented in network form is most often a combination of a number of smaller games, each representing individual decisions in a large decision space.

4. Game Action Types - the action type of a game determines how the physical procedure of how the game is played between the players. The major distinctions are as follows:

(a) Simultaneous Games - each play of the game, consisting of a strategic choice from each of the n players and the associated payoff, occurs at the same instant in time. This removes the ability for one player to learn about the strategies of the other without committing strategic information of their own.
CHAPTER 2. LITERATURE REVIEW

(b) Sequential Games - the plays made in these games occur at distinct points in time for each player in some predetermined order. In sequential games, a player can learn about another player’s strategic preferences by observing turns in which they themselves are not competing. This is especially critical in games of incomplete information, where each player is trying to construct a prior understanding of their opponent’s probabilistic strategy profiles.

(c) Repeated Games - repeated games are typically understood as a generalized case of simultaneous games in which the game is repeated more than once. In a sense repeated games blend behaviors of simultaneous and sequential games, in that each player must commit to strategy for each play but is still able to generate a prior distribution for their opponent throughout the course of multiple plays.

5. Bayesian Games - To define and elaborate on Bayesian games, we must follow the outline of Fudenberg and Tirole [31]. A player’s type encompasses any information that is not common knowledge but has an impact on that player’s strategic decision making process. In most circumstances, a player’s type is associated with their payoff functions, but this is not always the case.

6. Strict iterated dominance - for player $i$ possessing two strategies $A$ and $B$, $A$ dominates $B$ if player $i$’s payoff is greater when choosing $A$ than when choosing $B$, regardless of the strategy chosen by each player other than $i$. If there exists one scenario where the payoff of $A$ is greater than the payoff of $B$ and other scenarios have equivalent payoffs, $A$ is said to weakly dominate $B$. If the payoff of $A$ is always greater than the payoff of $B$, $A$ strictly dominates $B$.

7. Computational Aspects to the games - One of the major difficulties in working with computational games is determining how the equilibrium solution can be categorized (unique or not). Algorithmic game theory is a continually expanding field of research, but this dissertation only considers the requisite algorithms required to solve for equilibrium points in bimatrix games. This algorithm is known as the Lemke-Howson Algorithm [57].
CHAPTER 2. LITERATURE REVIEW

2-player zero-sum games can be solved for mixed equilibrium points using the minimax theorem and linear programming. 2-player games with different payoff tables for each player are known as Bimatrix Games, and equilibrium points can be solved using the Lemke-Howson Algorithm \[57\]. The Lemke-Howson algorithm tends to rise in complexity exponentially with the number of pure strategies. Other Methods include Simplicial Subdivision and Tracing Logit Equilibria. Evolutionary Game Theory has algorithmic merit, though Evolutionarily Stable Strategies must be researched. The mathematical restrictions are loose and likely will need to be tightened for application in industrial settings. Game Theoretic Strategies

2.1.1 Probabilistic Strategy Profiles

The Game Theory textbook from Fudenberg and Tirole outlines the definition of strategies. Something of interest to this research program is the concept of a Markov Strategy \[31\]. This could be done similar to the construction of a Markov Chain (including more information in a single state), but would tend to grow at a combinatorial rate. In 2010 Geoffrey Chamberlain published a view of strategy \[17\] that aimed to axiomatically treat strategy in a construct that could logically be built and compared to others. With that in mind, he presented a view of strategy based on 7 propositions: Chamberlains considera-

<table>
<thead>
<tr>
<th>Proposition 1</th>
<th>A strategy operates in a bounded domain, independent of tactical or operational considerations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposition 2</td>
<td>A strategy has a single, coherent focus.</td>
</tr>
<tr>
<td>Proposition 3</td>
<td>A strategy consists of a basic direction and a broad path.</td>
</tr>
<tr>
<td>Proposition 4</td>
<td>A strategy can be decomposed into distinct elements.</td>
</tr>
<tr>
<td>Proposition 5</td>
<td>Each individual component of the over-arching strategy is an intrinsically coherent concept relating to the basic direction of the strategy.</td>
</tr>
<tr>
<td>Proposition 6</td>
<td>The foci of a strategy imply the ability to influence a specific channel in the view of the firm.</td>
</tr>
<tr>
<td>Proposition 7</td>
<td>Each element of the strategy is formed either deliberately or emerges from firm considerations.</td>
</tr>
</tbody>
</table>

Table 2.1: Chamberlain’s 7 Propositions on Strategy
tions of strategy diverge from this research at this point, but one important consideration made is that a strategist possesses a cognitive bias that must be accounted for in the formation of any strategy. The cognitive bias theory is long-established in psychology literature \[49\], and is beyond the scope of this dissertation. The concept of mixed strategies has endured criticism \[8\] over the decades since its discovery for being problematic as a viewpoint, primarily from a mathematical and practical standpoint. A number of alternative views have been presented, starting with Harsanyi \[43\] and Rubinstein \[82\]. Rubinstein, in particular, comments on how random choices within a mixed strategy framework can be attributed to payoff-exogenous factors beyond the scope of the game. In response, Aumann and Brandenburger \[7\] present the idea that a mixed strategy profile is really in fact a profile of probabilistic beliefs about the strategies of the opposing players. One important consideration to those concepts of mixed strategy profiles is how to interpret the profile if the strategic requirement is to choose a single strategy for an instantiation of the game, and likewise required for estimating the strategy of a competitor. One major proposal of the research program is to understand what pieces of information would be most impactful in causing the strategic profile to collapse to a single decision almost surely.

2.1.2 Markov Strategies

A Markov strategy is one that depends only on the state variables of the game. For example a Markov Strategy applied in the game of checkers would only depend on the last move that an opponent makes, and does not consider multi-step strategic considerations. One important deviation from that will be the ability to include multiple components in a strategy. To include the restrictions of Markov Strategies and Markov Equilibrium points would be to limit the impact of important concepts such as reputation effects. Given the computational challenges involved in game theory, it follows intuitively that the Markov property would be attractive to simplify the problem.

2.1.3 Equilibrium Solutions

Equilibrium concepts in the preceding sections are generally refinements of the Nash equilibrium. A Nash equilibrium is defined as a solution concept in a game of \(n\) players
CHAPTER 2. LITERATURE REVIEW

in which no player has an incentive to deviate from their current strategies and in which players know all game and strategy information for the n-1 other players. Solutions having to do with mixed strategy profiles and Markov strategies are generalization on this concept. They can be divided as follows:

- Strong Nash equilibrium: In games of complete and perfect information, the Strong Nash equilibrium can be shown to exist.

- Subgame Perfect Nash Equilibrium: In games where a Strong Nash Equilibrium does not exist, a subset of the game may have an equilibrium point. This point is referred to as a Subgame Perfect Nash Equilibrium.

- Bayesian Nash equilibrium: In games of incomplete information [41], the Bayesian Nash equilibrium is defined such that each player possesses a strategy profile in which each strategy is a best expected response to that of all other opponents. In many games there tend to be many Bayesian Nash Equilibria.

2.2 Operations Research

Need to only consider the major methods from Operations Research that are being used in this dissertation:

2.2.1 Linear Programming

The theory of linear programming can trace its earliest history back to Fourier [90] in the 19th century. During the tumultuous period including World War I and World War II, the theory was improved even further by military analysts seeking to minimize the cost to a military supply chain. Kantorovich’s work [83] was key in this period, and subsequently Koopmans applied some of the same ideas to economics problems. Hitchcock [83] was next to contribute, providing an early version of what Dantzig would provide with the simplex algorithm [21]. The simplex method would stand as the primary algorithm to tackle linear problems until Karmarkar provided the concept of an interior-point algorithm [92]. Linear programming has found application in a vast number of fields since its inception.
CHAPTER 2. LITERATURE REVIEW

These include medicine, military science, economics, housing, social science, behavioral engineering, and board games.

2.2.2 Markov Chains

The field of Operations Research is vast, covering problem and solution concepts in both deterministic and stochastic domains. This research program will utilize methods in Linear and Nonlinear programming to optimally solve decision problems. Linear programming has found applicability in game theoretic analysis to a limited extent. Most game theoretic problems with 2 players can be solved using a simple minimax theorem. The minimax theorem allows for a solution as described above. A minimax result can take a number of forms, depending on the theorem being applied. But in general, the solution takes the following form, from [30]. Assume \( A \) and \( B \) are nonempty sets and \( f : A \times B \to R \) is a given function. A minimax result states the following:

\[
\max_{a \in A} \min_{b \in B} f(a, b) = \min_{b \in B} \max_{a \in A} f(a, b) \tag{2.2}
\]

If the min and max values are not attainable, inf and sup can be substituted into the above equation. After previous results were proven, a number of other minimax theorems surfaced to deal with ever-increasingly complex spaces. Results by Wald [100] expanded the generality and applicability of the collection of minimax theorems. However, application of the minimax theorem is limited at the moment to linear programming with 2 players. The formulation of the problem for 2 players is shown below, taken from the standard operations research text [44]:

\[
\begin{align*}
\max x_{m+1} \\
p_{11}x_1 + p_{12}x_2 + \cdots + p_{m1}x_m - x_{m+1} & \geq 0 \\
p_{21}x_1 + p_{22}x_2 + \cdots + p_{m2}x_m - x_{m+1} & \geq 0 \\
\vdots \\
p_{1n}x_1 + p_{2n}x_2 + \cdots + p_{mn}x_m - x_{m+1} & \geq 0 \\
x_1 + x_2 + \cdots + x_m & = 1
\end{align*}
\tag{2.3}
\]
CHAPTER 2. LITERATURE REVIEW

Where \((x_1, \ldots, x_m)\) are the probabilities of playing strategies \((1, \ldots, m)\). This problem solves the optimal strategy profile for player 1. The dual problem can be constructed in a similar fashion and represents the optimal probabilistic strategy profile for player 2. The variable \(x_{(m+1)}\) represents the minimax value of the game. A current open research problem is a modification of the minimax theorem for more than 2 players. Some other results have been established relating to the Minimax Theorem. Sion \cite{91} extended von Neumann’s concept of Minimax to nonlinear simplices where the only restrictions on the spaces are semicontinuity. Parthasarathy \cite{69} also made developments in the concept of a minimax theorem. Robust or stochastic programming techniques may allow for computation of more complex games in this domain. One major result in stochastic programming was the advent of two (or multi) stage programming formulations. Intuitively this can be thought of as a linear program contained within a larger linear program. The formulation given in Birge and Louveaux' textbook on Stochastic Programming is as follows:

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad c^T x + E [Q(x, \xi)] \\
\text{subject to:} & \quad Ax = b, x \geq 0
\end{align*} \tag{2.4}
\]

Where \(Q(x, \xi)\) is the optimal value of the second-stage program formulated as:

\[
\begin{align*}
\min_{y \in \mathbb{R}^m} & \quad q(\xi)^T y \\
\text{subject to:} & \quad T(\xi)x + W(\xi)y = h(\xi), y \geq 0
\end{align*} \tag{2.5}
\]

This formulation allows for optimal decision-making when two stages of decisions need to be made. The first stage involves some degree of uncertainty, and the choice made in the first stage directly attempts the decisions made in the second.

2.3 Graph Theory

In creating game-theoretic models with practical application, graph theory provides attractive computational properties. The concept of a graph enables an analyst to logically connect interrelated decisions and investigate decision impacts on both the large and small scales.
CHAPTER 2. LITERATURE REVIEW

2.3.1 History and Terminology

The first known application of graph theory can be found in Leonhard Eulers paper on the Seven Bridges of Konigsberg [10]. Eulers developments in this paper paved the way for the development of mathematical topology. The field continued to develop through the 18th and 19th centuries, and Denes Konig published the first definitive text on the subject in 1936 [51]. Frank Harary published what is now considered the world standard text in the field in 1969 [39], and provided a common communication medium for mathematicians and engineers. Of particular interest to this research is the study of probabilistic graphs, which were introduced as an analysis method by Paul Erdos and Alfred Renyi [25]. This method starts with n isolated nodes and sequentially adds random edges to the graph. The methods developed by Erdos and Renyi provide probabilistic bounds on when graphs with certain properties (namely, size, connectivity, and degree) will arise under this construction.

The study of graph theory lends itself to computationally understanding the connections between objects. The discrete construction of graphs allow for computation and have spurred the development of combinatorics as a field of study. It is helpful to provide some terminology to guide the reader in a discussion of graphs later in this proposal. A graph can be simply defined as an ordered pair \( G = (V, E) \) where \( V \) is the collection of vertices (or decisions in our scenario) and \( E \) is the collection of edges (creating feasible paths). Edges can be directed (only allowing movement from vertex A at the tail of the edge to vertex B at the head, and not vice versa) or undirected (allowing movement in both directions between any two vertices. Simple graphs (also called acyclic) are those graphs that do not contain loops. The order of a graph is the number of vertices contained therein. The size of a graph is denoted as the number of edges. Each vertex possesses a degree measure, defined as the number of edges that connect to it. The graphs discussed in this proposal will all be considered directed and simple, with finite order and size.

2.3.2 Computability

In transforming games from theoretical constructs to computable tools, the study of graphs and networks becomes critical. Various Algorithms exist for computing different
CHAPTER 2. LITERATURE REVIEW

aspects of a network, including shortest path (Dijkstra) and K-th Shortest Path (Eppstein [24], Aljazzar and Leue [2], or Yen [101]). These algorithms rely on data structures called heaps, specifically min-heaps, to enumerate the prospective paths, in sort in increasing order. An open research question, however, would be how to make this work in multiple dimensions (such as cost and performance). One cannot simply enumerate all the possible paths of a network, as the algorithms for such are known to be NP-complete.

2.3.3 Deterministic and Stochastic Formulation

The standard graph formulation is assumed to be deterministic, meaning that all vertices and edges in simple, directed graph G exist and allow directional flow. Each of the edges in the graph exist with probability 1. However, the subfield of stochastic graph theory has relaxed this assumption and provides another step towards a graph representation of real-world decision processes. A more directly applicable subset of random graph theory is known as the Network Probability Matrix, wherein entries of the matrix provide a probability $p_{ij}$ of whether the edge connecting vertex i to vertex j will exist in a given instantiation. This formulation was initially investigated by E.N. Gilbert in 1959 [34], and presented the concept of a stochastic graph as one central to practical problem in routing or information theory. This formulation also allows for the computation of strategic considerations that will be a cornerstone of this research program.

2.4 Mechanism Design

Mechanism Design was formalized as a field in the late 1960’s and early 1970’s. A number of Nobel laureates were involved in the development of Mechanism Design as a field, including William Vickrey, Leonid Hurwicz, Roger Myerson, and Eric Maskin. Other notable economists who have made contributions to the field of games with incomplete information include John Harsanyi and Reinhard Selten. The following section provides a brief history of Mechanism Design, tracing the development and major results from the field.

The major results in the theoretical foundations of Mechanism Design can be traced to the
work of economists in the 1970’s. Mechanism Design concerns itself mainly with a class of games in which multiple players rationally pursue their own objectives, each holding private information from the "designer" of the game. The first concept of a "mechanism" came from Leonid Hurwicz [6], who defined a class of games where players submit information to a "message center," with defined rules how the information is utilized in the game. The second major results came from William Vickrey [97], who devised a form of auction in which players holding private information could be induced to truthfully reveal their information to the auctioneer. This quickly led to a number of other results which broadened the applicability of the field beyond the so-called "Vickrey Auctions.”

Concurrently, the theory of games with incomplete information was being developed around this time by Harsanyi [40], who developed methods for analyzing games with players utilizing Bayesian probability rules. This work was critical in the development of Mechanism Design, as it laid the foundation for the formal concept of a mechanism that Leonid Hurwicz would propose. The following definitions are helpful in understanding the framework of a mechanism design problem:

**Definition 2.1.** A Type is a piece of private information held by a player in a game. For example, in an auction this could be the price the player is willing to pay, which he/she would not want to share with the other players.

**Definition 2.2.** A Solution Space is the feasible region in which game players can choose a strategy. Constraints like Individual Rationality and Incentive Compatibility help to confine the solution space.

**Definition 2.3.** A game exhibits the concept of Strategy-proofness when players cannot increase their utility by strategically misrepresenting their type. An example of this is the Vickrey Second Price Auction.

Building on the concept of a mechanism as defined by Hurwicz, Edward Clarke and Theodore Groves [20] [37] generalized Vickrey’s work and determined the mathematical environments in which strategies could be defined as strategy-proof, that is, environments which cannot be beneficially manipulated by the players. This concept would become known as Incentive Compatibility. However, a key separation still existed in the field:
CHAPTER 2. LITERATURE REVIEW

direct vs. indirect mechanisms. Indirect mechanisms, in which the designer places no restrictions on the connection between agent types and payoffs, occupy an infinitely large solution space. Direct mechanisms, in which the messages from the designer match each possible agent type, occupy a much more limited space. Allan Gibbard [33] showed that when a dominant strategy existed in the game, direct and indirect mechanisms were equivalent. Myerson [65], however, was able to show this result at the greatest level of generality. This meant that all indirect mechanisms under incentive compatibility could be characterized as direct mechanisms. This result is known as the Revelation Principle, which allows mechanism designers to only consider solutions which exhibit incentive compatibility. A number of results came in the 1980’s, but none approaching the scope of applicability of the Revelation Principle. Since its initial development, Mechanism Design has found applications in an increasing number of fields that deal with incomplete information. Initially, mechanism design was confined to economics, but since the late 1970’s, increasing efforts have been made to bring the field to other areas of research. These fields reach from social design [9], learning theory [5], to climatology [88]. The textbook by Fudenberg and Tirole [31] gives the most comprehensive overview of the field of Mechanism Design from a theoretical standpoint. The authors clearly lay out the theory, but the process of designing practical and applicable mechanisms is left to the reader. The process of designing mechanisms is examined from the standpoint of computer science in Nisan et. al [96](Nisan, Roughgarden, Tardos, and Vazirani, 2007). The lay out a number of algorithms which can be implemented to develop mechanisms for a number of applications, as well as give considerations to the complexity classes in which the algorithms fall. This last consideration is critical if mechanisms are to find applicability in fields where processing time is a limiting factor. Given the relative complexity of designing mechanisms that satisfy Incentive Compatibility and Individual Rationality, some fields resist direct application of Mechanism Design results and principles. The mechanisms themselves must be designed first, then tested to examine if they satisfy Individual Rationality and Incentive Compatibility.
2.5 Portfolio Optimization

The field of Portfolio Optimization has become the foremost tool in modern financial engineering. It defines the process of choosing an optimal allocation of securities in a financial portfolio, subject to some evaluation criterion. The field came about following the research of Harry Markowitz [59], who sought to maximize the portfolios expected rate of return with respect to some measure of portfolio risk. The relationship between risk and expected return allows financial engineers to generate the efficient frontier of a portfolio through a method called mean-variance analysis. The concept of efficiency in decision-making finally intersected with Operations Research techniques with the emergence of Data Envelopment Analysis (DEA), first published by Charnes, Cooper, and Rhodes [18].

2.5.1 Terminology and Computation

The fundamental concept of modern portfolio optimization is that assets in the portfolio should be selected to be globally optimal, that is, considering how the variation in each asset changes with the variation in every other [87]. This leads to the note that an investor makes a trade between risk level and expected return. Markowitz [59] defines the following quantities to relate to optimal portfolio selection: Expected Return is defined as:

\[
E(R_p) = \sum_i w_i E(R_i)
\]  

(2.6)

Where \( R_p \) is the return on the portfolio, \( R_i \) is the return on asset \( i \), and \( w_i \) is the proportion of asset \( i \) in the portfolio. The variance of portfolio return is defined as:

\[
\sigma^2_p = \sum_i w_i^2 \sigma_i^2 + \sum_{i \neq j} w_i w_j \sigma_i \sigma_j \rho_{ij}
\]  

(2.7)

Where \( \rho_{ij} \) is the correlation coefficient between the return on assets \( i \) and \( j \). Explicit formula constructions can be made for portfolio with \( n \) distinct asset classes. Of major concern to an investor seeking to maximize portfolio return for a given level of risk is how the component assets of the portfolio are correlated. Holding assets that are not positively correlated is one view of what economists call diversification [59]. The next major development in the field of portfolio optimization was the concept of an Efficient Frontier. When
CHAPTER 2. LITERATURE REVIEW

Possible portfolio holdings are plotted on the variance-return plane, Merton [61] showed that a hyperbolic region contains all feasible portfolio allocations.

This region is sometimes referred to as the Markowitz Bullet [94] and contains the individual securities that make up a portfolio. The region inside the bullet contains securities whose expected level of return for their given level of risk makes them a sensible addition to the portfolio. The upward-sloping portion of the curve is known as the Efficient Frontier and defines the optimal portfolio in terms of expected returns as a function of portfolio risk.

An additional topic to be addressed in this research is the application of portfolio theory to a company's technology portfolio, rather than a set of owned securities. This can take a number of forms ranging from corporate mergers and acquisitions to the acquisition of patent rights or intellectual property. The overarching need to be addressed by this research component is to understand how a company's technology portfolio can contribute to its business success, much in the same way as an investor's portfolio can contribute to his/her financial success. While financial securities can be purchased individually and generally function independently of one another (the key underpinning of diversified investing), technological acquisition are often highly interdependent.

For example, the implementation of a new semiconductor material in a computer chip may require additional software/hardware development to enable full use of the innovation. Individually, the semiconductor and the required software/hardware developments do not provide a proportional return on investment compared to the two innovations in tandem. The table below demonstrates this with notional benefits, which need to be further
defined. And while this example is highly notional, it demonstrates the idea that compo-

<table>
<thead>
<tr>
<th>Technology Innovation</th>
<th>Benefit to competitive position</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>New semiconductor material</td>
<td>10</td>
<td>Semiconductor not fully utilized without additional innovation</td>
</tr>
<tr>
<td>New software</td>
<td>6</td>
<td>New algorithms require additional support to be most effective</td>
</tr>
<tr>
<td>New hardware</td>
<td>5</td>
<td>Hardware required additional technology to reach potential</td>
</tr>
<tr>
<td>Combination of all 3</td>
<td>50</td>
<td>Combination benefit is synergistic, not simply a sum of all 3</td>
</tr>
</tbody>
</table>

Table 2.2: Technology Portfolio Components

ments of a technology portfolio cannot necessarily be considered as individual securities, complicating the application of modern portfolio theory. In order to bridge the gap between financial portfolio optimization and technology portfolio optimization, the theory by which securities are valued must be adapted to the valuation of technologies. This is not a straightforward process, but this document represents an introduction to the pertinent methods in financial engineering and security valuation.

The first major method is the Capital Asset Pricing Model (CAPM), described in many financial analysis texts, for example [38]. The CAPM is a model for pricing and individual security or capital asset, describing the appropriate rate of return for the return for the investment. This rate of return is then compared to another financial analysis method to determine whether the asset is an appropriate investment. If the actual asset price is greater than its CAPM valuation, the asset is considered undervalued, and conversely overvalued when the price is below the CAPM valuation. On a practical basis, the CAPM informs investors of the discounting they should expect in order to invest in an asset with a given level of risk. Intuitively, riskier assets should possess a greater discount. The CAPM equation is shown below:

\[
E(R_i) = R_f + \beta_i(E(R_m) - R_f) \tag{2.8}
\]
CHAPTER 2. LITERATURE REVIEW

Where \( E(R_i) \) is the return on the capital asset, \( R_f \) is the risk-free rate of interest, \( \beta_i \) is the sensitivity of excess asset returns to excess returns from the market, and \( E(R_m) \) is the expected return of the market. This model assumes that a portfolio can be optimized, with optimization being defined as the lowest level of risk for a given level of return. This relationship is often defined using the Markowitz Bullet in figure 2. A more complicated asset pricing model was published in 1973 by Fischer Black and Myron Scholes \[12\], becoming the aptly-named Black-Scholes model of asset pricing. The underlying equation for the model is a partial differential equation that describes the price of a European option (can only be exercised at the expiration date). The Black-Scholes formula is shown below:

\[
C(S,t) = N(d_1)S - N(d_2)Ke^{-r(T-t)}
\]

\[
d_1 = \frac{1}{\sigma\sqrt{T-t}} \left[ \ln \left( \frac{S}{K} \right) + \left( r + \frac{\sigma^2}{2} \right) (T-t) \right]
\]

\[
d_2 = d_1 - \sigma\sqrt{T-t}
\]

Where \( N(\bullet) \) is the appropriate Normal CDF, \( (T - t) \) is the time to maturity of the option, \( S \) is the spot price of the asset, \( K \) is the strike price, \( r \) is the risk-free rate of return, and \( \sigma \) is volatility of the asset returns. This model has a number of different formulations, and financial engineering literature contains versions of the model to accommodate American options as well (can be exercised at any time before the expiration). Black-Scholes tends to work well in practice, subject to a number of limitations. The normality assumptions of the model limit its sensitivity to extreme market behavior. In response, hedging strategies have been developed to reduce the overall risk to a firm.

2.5.2 Value at Risk

The concept of value at risk has proven effective in financial engineering theory and practice, and a good introduction is provided by \[46\] pp. 61-84. Another helpful resource in the theory of Value-at-Risk is found in \[45\]. The statement of a value-at-risk figure denotes some probabilistic information regarding the expected performance of an asset over a set period of time. Choudhry and Alexander \[19\] lay out the mathematical definitions for Value-at-Risk. Another related concept is conditional value-at-risk (CVaR), or expected shortfall. Various
authors note that \cite{45,46} notes that CVaR tends to be a conservative estimate of the risk associated with a portfolio. Some recent research associated with conditional value-at-risk is as follows:

- Perignon and Smith \cite{71} explore why financial institutions using proprietary Value-at-Risk models tend to be conservative in their estimations, resulting in excess regulatory funds. They conclude that the estimators used in the Value-at-Risk computations are biased.

- Goh, et al. \cite{35} investigate a new approach to conditional value-at-risk by partitioning the asset return distributions into positive and negative half-spaces. Their results improve on the classical Markowitz optimization program.

- Zymler, Kuhn, and Rustem \cite{103} provide a number of formulations to address the challenges of portfolio optimization when the portfolios contain derivatives. The authors prove that the formulations they provide represent coherent measures of risk, and connect well to the theory of robust optimization.

- Dias \cite{22} connects the theory of value-at-risk with market capitalization and shows it to be a relevant factor in the accuracy of the value-at-risk computations.

- Rockafellar, Royset, and Miranda \cite{81} develop a regression methodology that also provides a coherent risk measure. The authors apply this regression technique to problems in reliability engineering and financial risk management.

- Capinski \cite{15} provides a novel contribution by demonstrating how conditional value-at-risk computations for portfolios can be hedged using options. The paper is brief, but the concepts provides fertile connection to the topic of optimization.

2.6 Product Platforms

Product Platforms have revolutionized the way in which firms design products since their introduction in the mid-20th century. The textbooks by Meyer and Lehnerd \cite{62} and
CHAPTER 2. LITERATURE REVIEW

Ericsson and Erixon [26] provide a good overview on the topic. Meyer also provides one definition of a product platform:

"A product platform is a set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently developed and produced.” (Page 39)

These platforms have been applied across industries.

- Kokkolaras, Mourelatos, and Papalambros [50] introduce what they call a probabilistic version of analytic target cascading (ATC) to optimally design systems under decomposition. The authors’ goal it to introduce uncertainty into the design and model its propagation through a hierarchical system. While not explicitly referring to product platforms, the authors use of hierarchical decomposition and probabilistic inclusion bears consideration in this research. One area where my research differs from [50] is in firm-wide scope. The authors model optimal design considerations from a single decomposed product, whereas I look to create optimal subassemblies in order to generate a competitive product.

- Michalek et al. [63] bring in some important considerations that bear upon my research. The authors present a model that considers not only hierarchical subassembly optimization, but marketing and sales considerations as well. My research adds game-theoretic considerations to the model, specifically the expected competitive impacts of making a design choice vs. a competitor’s identical choice.

- Fixson [29] presents a literature overview of modularity and product commonality research from 1992-onward. Fixson notes that a sizable majority of the literature regarding product platforms refers to product design and manufacturing, with smaller percentages examining processes, organizations, and innovations. This dissertation focuses on products, but the firm portfolio scale.

- Farrell and Simpson [27] provide a mathematical model that aligns well with the business models I’m considering in this dissertation. The authors describe Product Platform Portfolios, which can be combined to meet customer needs in a competitive
CHAPTER 2. LITERATURE REVIEW

environment. One major assumption in this research is that target market segments are known *a priori*. I present a methodology later in the research for determining product platforms when the target markets and product requirements are not known with certainty.

- Zacharias and Yassine [102] point out that most modular product models are either too technical (engineering-centric) or not technical enough (marketing-centric). I agree with this assertion, and seek extend the research to include portfolio capabilities that are cutting-edge (new technology, etc.), rather than established materials and processes.

- Kumar, Chen, and Simpson [53] reiterate the point that very few design methodologies incorporate competitive considerations into their thought processes. This dissertation diverges from that of [53] in the causative responses that help create the product platforms. In Kumar’s paper, the platforms are designed in response to identified market niches, whereas this research presents a scenario in which changes are made to a product platform in anticipation of an upcoming product request or offering. This adds a good deal of uncertainty to the problem.

- Shiau and Michalek [85] conclude that if a consumer’s preferences exhibit heterogeneity, than the optimal design for a product is coupled with market preferences. Note that the models presented in [85] do not possess a bi-directional interaction between customers and providers, which is a modification present in the research presented here.

- Consequently, Shiau and Michalek [86] present a more complex model in which Nash and Stackelberg (leader-follower) conditions are imposed on the game. The authors show that considering competitor reactions and interactions improves the profit margins for all products involved. More importantly, they demonstrate that ignoring the actions of competitors in the space results in drastic overestimation of profits.
CHAPTER 2. LITERATURE REVIEW

- Luo [58] presents a computational study on how the marketing process should be integrated into optimal engineering design.

- Qu et al. [77] present a genetic algorithm for developing an optimal product design. In doing so they propose using a graph-theoretic representation of a product, to which I will later apply game-theoretic analyses. The authors refer to this as an "embryonic product platform" and utilize the evolutionary aspects of generic algorithms to demonstrate the development of an effective platform. I consider several of the identified areas for future research drawn from this paper:
  - The addition of game-theoretic considerations.
  - The mathematical formulation of a product family.

- Fujita, Amaya, and Akai [32] approach the optimization problem from a global perspective, tying in supply chain considerations as well. They utilize genetic algorithms along with mixed-integer programming methods to develop optimal designs for what they call a "Global Product Family." I alter one fundamental assumption in this dissertation: the customer in the competition is attempting to solicit optimal designs from possible contractors and have direct interactions therein, rather then the contractors developing designs independent of interactions with the customer base.

- Alsawalqah, Kang, and Lee [3] approached a different question related to the formulation of product platforms: the scope of the platform. While they authors utilized software as a testbed for this approach, the pertinence of the question is critical for the application of product platforms in this dissertation.

- Ramadan and ElMaraghy [78] provide methods of measuring the diversification among platforms contained within a single firm. This is important if a product platform methodology is going to be implemented on a broad scale, as platform should be distinct enough to justify the investment in each of the platforms considered.

- Along the line of thinking taken by Ramadan and ElMaraghy, Schuh, Rudolf, and Vogels [84] present mathematical models to measure the performance of product
platforms implemented in a market space. The authors also provide alignment between product platform effectiveness measurement and corporate objectives.

One of the major questions that needs to be addressed going forward is the causality of product platforms, investment, and competitive product offerings. Do customers present product requests based on the technologies or capabilities that they observe in a firm, or do firms invest in technologies or capabilities in response to customer demands.

### 2.7 Customer Relationship Management

The field of customer relationship management (CRM) spans a large number of fields, but some conclusions from recent CRM literature bear consideration in this research. When considering competitive business decisions that involve imperfect information, a fertile avenue for improving the information deficit from the customer is when a firm actually engages with the customer in discussion. Most research in the field of CRM is based in the service industries, but the articles detailed below demonstrate that principles can apply in competitive situations as well. The academic literature refers to this as Voice of the Customer research, and some recent results demonstrate what is currently known about the concept.

- Bove and Robertson [13] conducted a large study examining whether customers in a service industry were likely to voice feedback to the service provider. While the process may seem commonplace, the authors found that customers voiced feedback and received redress when they believed their feedback would be acted upon. Additionally, the authors also note that customers frequently voice feedback in the form of negative word-of-mouth when they choose not to voice the same feedback directly to the organization. This speaks to the importance of reputation and how a lack of response can indicate a serious shortcoming in terms of perceived capability.

- Netzer, Lattin, and Srinivasan [66] take a technical approach to analyzing the relationships between customers and providers. The authors utilize a Hidden Markov Model (HMM) to examine giving patterns. This model allows the authors to predict future giving patterns based on a number of unobserved states. The authors’
acknowledgement of imperfect information in the relationship between customers and firms lends credence to this body of research. The classification of unobserved states in the model can be applied across industries, though the collection of data may prove challenging in some circumstances.

- McColl-Kennedy, Sparks, and Nguyen [60] examine the impact of a poor service encounter on the relationship between a customer and a service provider. The authors conducted a large statistical study which examined when customers would direct anger at an organization versus a specific individual. While I am not directly examining customer emotions, this paper is helpful in understanding retinence in collecting imperfect information.

- Lacey [54] examines how a customer voice affects marketing impact. Using a structural equation model with a large sample size in two distinct industries, Lacey demonstrates that allowing customers to have a voice leads to increased sales volume and repeat customer base. Similar to [13], this article highlights the importance of reputation in competition under imperfect information.

2.8 Summary of Literature Review

As evidenced by the preceding sections, this dissertation combines concepts and computational algorithms from a number of fields. This fusion reflects the complexity of decision-making under imperfect information. Each field reviewed in this chapter has strengths and limitations that curtail applicability. However, it is the intent of this research to develop connections between previously disparate fields of study. Doing so will allow firms to construct strategies that provide consistent and sustainable competitive advantage.
Chapter 3

Winning Business on the Large Scale

3.1 Firm Decisions

One major impetus for this chapter is to integrate concepts from Product Platform development, competitive analysis, and operations research. As discussed in the literature review, product platforms have informed the design and production of a huge number of successful designs across a broad swath of industries. Competitive analysis techniques are traditionally applied at a high level, but this chapter seeks to enable decision-making at all organization levels. This chapter marks a departure from the standard paradigm of product platform research. A good commentary with the standard terminology can be found in both Meyer [62] and Ericsson [26]. The key thought that distinguishes the models found in this chapter from previous research is the causality relationship found in the product development phase.

Product platforms have found standard applications in commercial products whose uses are defined a priori and whose targets market spaces are defined before the release of the product. However, when the need for a product is not defined as a known market space and only released as a request for proposal, the applicability of a product platform changes. The return for platform utilization can no longer be calculated in the standard form, as products based on the platform may not be produced in mass quantities. Consequently, a product platform developed in this scenario should be adaptable to future design requirements that cannot be anticipated with certainty.
CHAPTER 3. WINNING BUSINESS ON THE LARGE SCALE

A product platform $P$ is defined as a collection of components and capabilities

$$t = (t_1, \ldots, t_n)$$

Now considering the technology portfolio for the firm, which consists of available product platforms as well as other capabilities ($c_i$) not categorized into platforms:

$$T = (t_1, \ldots, t_n, c_1, \ldots, c_m)$$

Through acquisition, divestiture, or internal research and development, the components of the portfolio will change to support different business needs and pursuits. In trying to measure and optimize the size and scope of the product platforms available to a firm, we must understand the process by which requests are made of the firm that require the utilization of the technology portfolio. The flow and decision process is shown in figure 3.1:

Figure 3.1: Schematic detailing the interaction of corporate and competitive strategy.

Let the arrival process of requests be a stochastic process as follows:

$$R = (r_1, r_2, \ldots)$$

(3.1)

Also let each request $r_i \in R$ map to the elements in a firm’s technology portfolio by some function

$$G = \{t_1 \rightarrow r_i, t_2 \rightarrow r_i, \ldots, t_n \rightarrow r_i\}$$

(3.2)
Each request \( r_i \) has a profit value distributed according to some distribution \( F \). Intuitively, a high value of mapping \( G \) would indicate that a firm is well-aligned to a product request. This follows from the fact that each component of the technology portfolio maps to some requirement or aspect of the product to be developed. A major question to answer is how to alter the available product platforms through investment, acquisition, or divestiture such that the long-term return of investment is maximized. In order to set up the optimization problem, a number of quantities must be defined. The current level of return without any additional alterations is defined as \( \rho^* \). Alterations to the product platforms will change the rate of return for the firm by augmenting the likelihood of winning any \( r_i \in R \).

Firms approach this model of business using a combination of product platforms and individual engineering, manufacturing, and development efforts. This leaves a firm with three major options:

1. Augment current product platforms to increase or decrease the functional scope
2. Divest the functionality in a current product platform into individual functional components
3. Combine functional elements currently produced individually into a new product platform

The remainder of this chapter provides quantitative and qualitative models to address these three options.
3.2 **Definition of Strategies**

Intuitively, product development and release in a competitive market space requires a large number of interrelated decisions. Ideally these decisions coalesce to form a competitive product that will perform well in sales. However, the number of decisions made in preparation for competitive product release have the potential to generate a combinatorially large number of possible strategies. Some of those strategies may be dominated by others and consequently eliminated, but the potential for enormous normal form games exists. So naturally the questions arise of how to handle a combinatorially large number of possible strategies. One possible technique is to reduce the number of competitive considerations so as to keep the number of strategies tractable. However, this method has the potential to remove strategies that are optimal and form the core of the game. Alternatively, this chapter demonstrates a different alternative to handling large numbers of competitive strategies. The network formulation of a game presented in 6 allows us to make a relative determination of how the components of a strategy interact with one another. Large firms with multiple divisions and programs produce a broad variety of products and services. These firms often utilize past experience in product development and manufacturing to position themselves strategically for upcoming pursuits. It would benefit a competitive intelligence group to understand two things:

1. The network construction of a firm’s own product portfolio and how technical and production expertise have positioned the firm for future pursuits.

2. The possible network construction of a competitors product portfolio.

For consumer products, this information can be gleaned from actually purchasing the product and reverse engineering any technology included therein. For large installations or products with proprietary technology, developing this network is more difficult and may require probabilistic network models. A robust understanding of both of these models would provide information otherwise unobtainable. For example, a quantifiable description of critical technologies, that is, those capabilities which influence and enable a large number of programs and pursuits. Correspondingly, a sense of the critical technologies seen by a competing firm. When considering a new business pursuit, knowledge of the strengths
and weaknesses of both the firm and other competing firms will shape what changes or investments need to be made in order to make a firm competitive for a given pursuit.

3.3 Technology Portfolio Network

This chapter introduces the concept of a Product Portfolio Network, which considers all items or services produced by a firm and connects them via technological or other linkage. The product portfolio network is comprised of several components:

1. A collection of technological capabilities on which the firm can draw to design a new product.
2. A set of requirements on which a new product will be based

A given product will often not require the input of all the technological capabilities possessed by the firm. Rather, a firm will efficiently utilize their technological strengths to design a product for the required levels of cost and performance. In the case of analyst developing a Product Portfolio Network for their own firm, it is likely that the analyst will have access to a great deal of information, though proprietary, with which to construct the network for the firm. When the same analyst is tasked with developing the product portfolio network for a competing firm, the problem of incomplete information becomes apparent. The analyst does not have access to any proprietary information from the competing firm, and must therefore generate a network based on inference or other probabilistic methods. If a firm has production or marketplace history, a competitor can potentially construct the technology portfolio network based on that information. One important note is that the values of the linkages between elements in the technology portfolio will now be stochastic rather than deterministic.

3.4 Optimization of Investment

Assuming a product portfolio network exists, the information should be used to inform business pursuit decisions. How can this information be used to develop optimal investment
decisions? In most firms, the time spent in developing proposals for new business pursuits is internally funded, and consequently the money spent on those efforts must be carefully apportioned. Given that any money invested in proposal efforts cannot guarantee contract wins, optimization of investment decisions must be computed in terms of expected value or stochastic models.

### 3.4.1 Capability Optimization

Here I present an optimization model for purchasing and/or divesting technical capabilities in a competitive scenario. I begin by defining the following:

- $C^+ =$ technical capability currently contained in a firm’s portfolio
- $C^- =$ technical capability currently not contained in a firm’s portfolio
- $\alpha_i =$ cost associated with acquiring or divesting capability $i$
- $J =$ interaction matrix detailing how two capabilities impact each other
- $Y =$ filtering matrix to show which capabilities currently reside in the portfolio
- $V =$ binary vector acting as a portfolio starting point
- $x_i =$ binary decision variable for acquiring/divesting capability $i$
- $A =$ total available investment budget

Several of the above quantities bear further explanation. $V$ is necessary to define because a firm will likely not be able to develop a technology portfolio from scratch. Consequently, the firm may need to divest some capabilities and invest in others. This requires a mathematical representation of the technological starting point. The simple version of this problem assumes that all available competitions that could be accessed by acquiring or divesting capabilities are known at the outset, and the decision variables represent an effort to engage in the set of competitions with the highest computed payoff.

\[
\begin{align*}
\text{max } & \quad (X \ast Y + V) \ast J \ast C \ast (X \ast Y + V)' \\
\text{s.t.} & \quad [X \ast Y + V] \alpha \leq A
\end{align*}
\]
CHAPTER 3. WINNING BUSINESS ON THE LARGE SCALE

This program is a quadratic binary integer program, as can be seen from the objective function. One point of note is that decision variable \( x_i \) can only take on values of \((0, 1)\), where a value of 1 indicates a change in the capability. This is taken to mean to divestiture of a capability from the current portfolio or the addition a new capability from outside the current portfolio.

3.4.1.1 Capability Computational Example

To illustrate, here I provide a small computational example:

\[
C = (c_1, c_2, c_3, c_4)
\]
\[
C^+ = (c_1, c_2)
\]
\[
C^- = (c_3, c_4)
\]
\[
V = (1, 1, 0, 0)
\]
\[
Y = \begin{bmatrix}
-1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
\[
J = \begin{bmatrix}
0 & 0 & 1.5 & 0.6 \\
0 & 0 & 2 & 1.1 \\
1.5 & 2 & 0 & 1 \\
0.6 & 1.1 & 1 & 0
\end{bmatrix}
\]
\[
C = \begin{bmatrix}
2 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 \\
0 & 0 & 6 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
\[
A = 4
\]
\[
\alpha = [1, 1, 3, 2]
\]
CHAPTER 3. WINNING BUSINESS ON THE LARGE SCALE

Solving the quadratic binary optimization program in §3.3 yields the following:

\[ X^* = (1, 0, 1, 0) \]
\[ Z = 18 \]

The solution indicates that we should make a change in status for capabilities 1 and 3 (divest #1 and acquire #3). This yields an objective function value of 18 (compared to the previous value of 5). By Inspection we can verify this to be the optimal solution due to the relatively small number of feasible solutions in this problem.

### 3.4.2 Investment Optimization

The previous problem dealt with the decision of what capabilities to include in a firm’s portfolio, whether acquired or divested. The following optimization problem deals with a similar problem: how to optimize research and development investment in the capabilities already present in a firm. If the capabilities of the firm are currently known to be optimal or change to the portfolio is not currently possible, the following program can be applied:

\[ X = (x_1, \ldots, x_n) \]

\( M = \) matrix defining unit increase in contribution for each element \( x \in X \)

\( J = \) interaction matrix where \( J_{i,k} \) defines the scalar change for each pairwise combination

\( B = \) maximum investment level for each \( x_i \in X \)

\( A = \) total available investment amount

This allows us to define the following optimization problem:

\[
\begin{align*}
\max Z &= \sqrt{X} \cdot M \cdot J \cdot \sqrt{X}' \\
\text{s.t. } X \cdot J &\leq b \\
\sum X &\leq A
\end{align*}
\]

(3.4)
CHAPTER 3. WINNING BUSINESS ON THE LARGE SCALE

3.4.2.1 Computational Example

Again, I illustrate this model with a straightforward computational example:

\[ X = (x_1, x_2, x_3) \]

\[ M = \begin{bmatrix}
1.5 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 1.8
\end{bmatrix} \]

\[ J = \begin{bmatrix}
0 & 0.6 & 1.2 \\
0.6 & 0 & 1.5 \\
1.2 & 1.5 & 0
\end{bmatrix} \]

\[ A = 5 \]

\[ B = (1, 2, 3)' \]

Solving the above program yields the following:

\[ X^* = (0.651, 2, 2.349) \]

\[ Z^* = 48.43 \]

After solving the problem, we notice that the constraint associated with \( x_2 \) is binding, indicating that no additional capital can be invested without reducing the value of the objective function. It is still an open problem whether the two optimization programs presented herein can be combined into a single decision program.

Based on the two optimization programs given in this section, two different questions are answered:

- The first program informs an analyst which capabilities should be acquired or divested.

- The second program examines the current elements in a technology portfolio and how they can be optimally invested based on their individual and pooled responses.

The combination of the two methods is important for a firm seeking to apply this to their portfolio. Combined effects from more than 2 capabilities are possible, but the resulting optimization program is an area of future research.
CHAPTER 3. WINNING BUSINESS ON THE LARGE SCALE

3.5 Corporate and Competitive Strategy Computation

Based on Porter [73], I propose a new computational framework for computing optimal firm structure. One critical aspect to the formulation is making the distinction between corporate and competitive strategies for a firm.

"Competitive strategy concerns how to create competitive advantage in each of the businesses in which a company operates. Corporate strategy concerns two different questions: what businesses the corporation should be in and how the corporate office should manage the array of business units."

In this formulation there are multiple decisions being undertaken at different levels of the organization. First, decisions at the top levels of the firm are made with how each of \( n \) business units should be allocated (or positioned) in a market space. Second, the competitive strategy in each of the business units is unique and must be computed as well.

3.5.1 Market Spaces

This formulation required a mathematical description of both the firm in question and the marketplaces in which the firm operates. We define a marketplace \( M \) as a collection of individual market spaces \( m \in M \), which may or may not overlap. Each market space has some current value \( c \in C \), which denotes the amount of money paid to a winning competitor for controlling the entire space. The total amount of money available is \( C \). Intuitively, some market spaces in which a firm competes will overlap. For example, Samsung produces both televisions and computer displays. Due to a similar technological foundation, activities and linkages in one space may provide benefits in the other. Define the level of commonality between two market spaces as \( \phi_{ij} \in [0, 1] \). For the \( M \) market spaces being considered, define a matrix \( \Phi \) as follows:

\[
\Phi = \begin{bmatrix}
0 & \phi_{12} & \cdots & \phi_{1m} \\
\phi_{21} & 0 & \cdots & \phi_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\phi_{m1} & \phi_{m2} & \cdots & 0
\end{bmatrix}
\]  

(3.5)
CHAPTER 3. WINNING BUSINESS ON THE LARGE SCALE

One possible method of computing $\phi_{ij}$ is to compute the cost of the components and manufacturing methods that products in connected spaces share as a fraction of the overall production cost of each product.

3.5.2 Relationship to Product Platforms

A number of approaches have been developed for improving the operational efficiency at firms with heterogeneous product development and production. *Product platforms* are defined by Robertson and Ulrich [80] as follows:

"By sharing components and production processes across a platform of products, companies can develop differentiated products efficiently, increase the flexibility and responsiveness of their manufacturing processes, and take market share away from competitors that develop only one product at a time."

The authors mentioned a number of benefits to be gained through the implementation of product platforms. Product customization, reduced manufacturing cost, and lower risk are all expected benefits of implementing product platforms. We utilize four product platform relationships from [80] to formulate potential network nodes for this analysis.

1. Components
2. Processes
3. Knowledge
4. People and relationships

Product Platform Planning, as defined by the above authors, provides a number of improvements in operational effectiveness. However, as Porter [75] notes, that "operational effectiveness is not strategy." Porter rightly notes that improvements in operational effectiveness cannot be sustained forever, as best practices will rapidly be diffused throughout an industry.
CHAPTER 3. WINNING BUSINESS ON THE LARGE SCALE

3.5.3 Production Strategy Networks

Accompanying the mathematical description of the appropriate market spaces and their interactions is the set of processes undertaken by the firm to develop a product in that space. We define the aforementioned network as a collection of activities (components and processes) and the linkages between them.

- One initial formulation of the production strategy network is based on the product bill of materials (BOM), each item of which is integrated through a production process to form a product.

- Subassemblies of the product represent their own graph components and are connected to others via the assembly process.

- This formulation allows to develop an estimate of how investment impacts the product as a whole, and whether that investment will filter through to other products.

The development of this network is critical for a computational understanding of competitive and corporate strategy, and even a Bayesian estimation of the network for a competitor is of value in estimating likely interdictions.

3.5.4 Small Example

Consider the following example of a firm that competes in 2 market spaces. Each of the spaces represents an identified market segment in which the firm produces a product. The overlap between the spaces indicates that the 2 products produced share some common components or manufacturing processes. The firms possesses a competitive strategy for each of the products being sold, represented by a collection of activities and linkages. Note that 1 of the activities is shared between the product networks, indicative of the overlap between spaces. Maintaining separation between every market space in which company competes allows for a firm to develop a highly customized product for the market space, with an individual set of activities and linkages. However, this separation comes at the expense of increased and possible repeated production costs. On the other hand, increasing the level of overlap will decrease the production costs (driving towards mass production)
CHAPTER 3. WINNING BUSINESS ON THE LARGE SCALE

Figure 3.3: Two Market spaces in which a firm sells products at the expense of being competitive in distinct market spaces.

This formulation follows closely with the thought process of Porter [72], and we can clearly delineate competitive strategy (the set of activities and linkages for each product or business unit) and corporate strategy (the relative positioning of each product or business unit). The optimization problem to be formulated here involves two set of interrelated decisions: those made by the business unit leadership and those made by the firm leadership. The decisions are interrelated, as a choice by firm leadership to more closely integrate the activities of two business units directly impacts the strategies chosen by the respective business unit leaders. We assume a binary result for each of the market spaces in which the firm competes, meaning winning the competition in space $i$ results in a payment of $c_i$ to the firm. Partial winning or sales are not allowed.

3.6 External or Environmental Factors

One consideration made by Porter [74] is that a comprehensive theory of strategy must do 2 things:

"First, a theory must deal simultaneously with both the firm itself as well as the industry and broader environment in which it operates. The environment both constrains and influences outcomes, which the more introspective resource view neglects. Second, a theory must allow centrally for exogenous change, in areas such as buyer needs, technology, and input markets."

The formulation presented in the previous section addresses the internal characteristics of the firm and the decisions made in terms of competitive position. However, to be comprehensive we must address exogenous factors that can impact both competitive and corporate
strategy. Additional model components are needed to compute the impact of exogenous changes in the environment.

### 3.7 Closing Remarks

This chapter presents a number of optimization models, both quantitative and qualitative, that relate to how a firm structures its capabilities in order to win business. These models provide substantial new computational capabilities. However, the research area is still new, and additional model development can be done to make these models applicable across a broad range of applications.
CHAPTER 3. WINNING BUSINESS ON THE LARGE SCALE

Figure 3.5: Firm decision to separate two market spaces

Figure 3.6: Firm decision to further integrate two market spaces
Chapter 4

A Higher-Order Markov Property

4.1 Introduction to Markov Processes

In the theory of stochastic processes, Markov Processes are those that possess the Markov Property. A standard text in probability theory can provide the necessary foundation, and I refer the reader to [23] from which the descriptions below are taken. The general case of the Markov Property is defined as follows. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with a filtration $(\mathcal{F}_t, t \in T)$ for some index set $T$, and let $(\mathcal{S}, \mathcal{S})$ be a measure space. An $\mathcal{S}$-valued stochastic process $X = (X_t, t \in T)$ adapted to the filtration is said to possess the Markov Property with respect to the filtration $\mathcal{F}_t$ if, for each $A \in \mathcal{S}$ and each $s, t \in T$ with $s < t$,

$$
\mathbb{P}(X_t \in A | \mathcal{F}_s) = \mathbb{P}(X_t \in A | X_s)
$$

(4.1)

If the process being considered occurs with a discrete set $S$, a discrete sigma algebra, and $T = \mathbb{N}$, the equation above can be reformulated into the more commonly encountered description of the property:

$$
\mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_0 = x_0) = \mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1})
$$

(4.2)

The Markov property states that the probability of a process being in some state in the next time period only depends (in the discrete case) on the current state of the system. This
allows processes possessing the Markov Property to disregard any probability information occurring before the current time period.

4.2 Importance in the Development of Stochastic Processes

Given the challenge of considering processes with complex histories, the Markov property provides a simplification in making the problems tractable. We will give a number of examples of stochastic processes whose existence and applicability are owed to the Markov Property. Markov Chains, in both discrete and continuous time, depend on the Markov Property for the development of methods to analyze the models. The Chapman-Kolmogorov (from [44] equations define the probability of state transitions in stochastic processes possessing the Markovian property.

\[
p^{(n)}_{ij} = \sum_{k=0}^{M} p^{m}_{ik} p^{n-m}_{kj}
\]

for all \( i \in 0, \ldots, M, j \in 0, \ldots, M \) and any \( m \in 1, 2, \ldots, n - 1 \) and \( n = m + 1, m + 2, \ldots \)

The Chapman-Kolmogorov equations allow us to compute the likelihood of ending up in any state \( i \) \( n \) steps after starting in state \( j \). Allowing \( n \to \infty \) yields what we call the steady-state distribution of the system. The steady-state distribution can be computed in several ways. First, the following simultaneous equations can be used:

\[
\pi_j = \sum_{i=0}^{M} p_{i} p_{ij}
\]

\[
\sum_{i=0}^{M} \pi_i = 1
\]

Alternatively, the following matrix form can be used:

\[
\Pi = (1, \ldots, 1) \ast (I - P + ONE)^{-1}
\]

where \( P \) = \( (\pi_1, \ldots, \pi_M) \), and \( ONE \) = an \( M \times M \) matrix of ones. The shortcoming to the second method is the challenge of computing the matrix inverse for Markov chains with a large state space.
4.3 Limitations in Stochastic Model Usage

Using the Markov Property to construct a stochastic model presents a number of challenges to an analyst. First, the analyst must show that historical transitions in the process depend only on the current state to within a certain level of statistical confidence. This investigation could involve tests for autocorrelation or other higher-order characteristics. Second, an analyst must understand the underlying dynamics of a stochastic process must be well understood. For instance, is the assumption that history beyond a current state reasonable for a given application? For example, the state space of some process may grow or shrink in response to exogenous influences. Consequently, the only choices for constructing the stochastic model are to a priori include all possible states in the Markov Chain (which for some processes can be intractably large), or to abandon the Markov Property and construct a model with different probabilistic properties. An alternative method exists for including additional information in each state variable of the Markov Chain. Rather than define the stochastic states as single realization of an indexed random variable, define the process state as a collection of consecutively indexed random variables. This serves to allow the construction of models that include some form of "history," but still retain the computational benefits of the Markov Property. On the other hand, the state space now increases to include all combinations of states in the consecutive indexing. An example of two Markov Chains is now given for clarity. The first chain possesses single-factor states, while the second chain possesses states with multiple components. For state space \( S \), every state \( s_i \) is described as the realization of a stochastic process at time \( t \).

\[
P = \begin{bmatrix}
  s_1 & s_2 & \cdots & s_n \\
  s_1 & p_{11} & p_{12} & \cdots & p_{1n} \\
  s_2 & p_{21} & p_{22} & \cdots & p_{2n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  s_n & p_{n1} & p_{n2} & \cdots & p_{nn}
\end{bmatrix}
\] (4.6)

Note that in (4.6) the state space \( (s_1, s_2, \ldots, s_n) \in \mathbb{R}^1 \). Contrast this Markov Chain with the following, where the state space consists of states that are function of multiple time epochs. The following Markov Chain details the state descriptions when the state is a function of the
current as well as one epoch previous. Consequently we need to redefine what constitutes a state $S^n$:

$$S^n = (X_{t-n}, X_{t-n+1}, \ldots, X_t)$$ (4.7)

When considering a Markov Chain in which the state definition consists of more than one time epoch, one key distinction to make is that the probabilities cannot simply be the product of the single-step probabilities. If this is the case, the chain is organized as a block matrix and the long-run properties are unchanged. Consequently, the inclusion of historical information was of no additional value.

### 4.4 Results from Computational Linguistics

The phenomenon of multiple components of information being included in a state of a Markov Chain has been encountered before. The field of computational linguistics terms the phenomenon an *n-gram*. N-grams are then used as predictors in natural language processing, taking a corpus of information and using it to predict upcoming words or phrases [14, 28, 89]. Another set of application of n-grams comes from computational biology, where n-grams of gene sequences are used to predict follow-on gene sequences [64, 93, 98]. Another biological application is making probabilistic estimates of the species associated with a given genetic sample based on detected partial sequences.

The authors above note several problems that must be overcome when considering n-gram sequences in any time series:

- The *zero-frequency problem* occurs when a given n-gram does not occur in a time series. Consequently, any probabilistic prediction on the future likelihood of occurrence will be 0 as well. A number of methods have been proposed for accommodating this phenomenon:
  - The “add-one” method, also referred to as Laplace Smoothing
  - Good-Turing Discounting
  - More sophisticated methods involving Bayesian priors and posteriors.
CHAPTER 4. A HIGHER-ORDER MARKOV PROPERTY

In this research we will utilize simple Laplace Smoothing to accommodate missing elements in the transition probability matrix.

To my knowledge, this research represents the first application of n-grams to business development and strategy problems. In contrast to previous applications in the biological sciences and natural language processing, I apply the n-gram concept to detecting higher-level pattern detection and Markov Analysis. Specifically, I aim to answer the following questions regarding the market share of a firm:

- Starting with a market share of \( p \in [0, 1] \), what is the eventual steady state distribution of market share?

- Starting with a market share of \( p \), what is the expected time until the firm reaches some other market share \( q \neq p \)?

In formulating this Markov Chain, we assume that the transitions to states are governed solely by the transition probability matrix and are insensitive to external factors such as governmental intervention and regulation.

The probability question being answered in this section is as follows: Define \( B(S) \) as the collection of n-gram states in \( S^n \) whose first entry we are interested, say \( i \). Also define \( E(S) \) as the collection of n-gram states in \( S^n \) whose last entry is \( j \). We are seeking to compute the expected transition from any \( b \in B(S) \) to any \( e \in E(S) \).

The first possibility is that a state \( s \in B(S) \) contains \( j \). Entrance into this state indicates that the process will transition from state \( i \) to state \( j \) in less steps than the cardinality of the n-gram state, i.e. the desired destination already exists in the initial n-gram state. The second possibility is that the desired destination \( j \) does not presently exist in \( b \). Consequently, the process will have to transition to an intermediate state that eventually contains \( j \).

A number of elements are needed to compute the expected transition times between atomic states in the state space. When considering a Markov process at an n-gram level (\( n \in (1, \ldots, \infty) \)), define a series of transition probability matrices \( \mathbb{P} = (1^P, 2^P, \ldots, n^P) \). \( 1^P \) is equivalent to the standard 1-step transition probability matrix for a Markov chain. \( 2^P \) is the transition probability matrix when considering 2-gram states. We illustrate this using the
following simple example:

\[ P = \begin{bmatrix} s_1 & s_2 \\ s_1 & \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} & s_2 \end{bmatrix} \]  

(4.8)

The Markov chain in 4.8 is the standard Markov chain considering only stochastic transitions between single state elements. We now move to \( ^2P \), below (we assume all states are indexed on \((1, \ldots, n)\):

\[ P = \begin{bmatrix} (s_1) & (s_1) & (s_2) & (s_2) \\ (s_1) & p_{11} & p_{12} & p_{13} & p_{14} \\ (s_1) & p_{21} & p_{22} & p_{23} & p_{24} \\ (s_2) & p_{31} & p_{32} & p_{33} & P_{34} \\ (s_2) & p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \]  

(4.9)

Transition probability matrices \( ^3P, \ldots, ^nP \) can be similarly calculated.

Computational example:

- \( S = (S, C, R) \)
- \( S^n \) = all \( n \)-wise combinations of the elements in \( S \)
  - \( S^2 = (SS, SC, SR, CS, CC, CR, RS, RC, RR) \)
- If \( s_1 = S \), \( B(S) = \{SS, SC, SR\} \)
- If \( s_1 = S \), \( E(S) = \{SS, CS, RS\} \)
CHAPTER 4. A HIGHER-ORDER MARKOV PROPERTY

4.5 Notional Computational Example

Consider the following discrete-time Markov chain:

\[
P = \begin{bmatrix}
1 & 2 \\
1 & 0.6 & 0.4 \\
2 & 0.3 & 0.7
\end{bmatrix}
\] (4.10)

Computation of the long-run characteristic of the chain are straightforward. The steady state distribution \(\Pi\) and collection of mean first passage times are shown below:

\[\Pi = (0.4286, 0.5714)\]

The following matrix is the collection of mean first passage times, where entry \((i, j) = \mu_{ij}\)

\[
\begin{bmatrix}
1 & 2 \\
1 & 2.333 & 2.500 \\
2 & 3.333 & 1.750
\end{bmatrix}
\]

Now consider the following discrete-time Markov chain where using a 2-gram state. The transition probabilities below are identical to the 1-gram chain, which assumes that higher-order effects (those that would violate the Markovian property) are not significant.

\[
P = \begin{bmatrix}
1 & 2 \\
1 & 1 & 0.6 & 0.4 & 0 & 0 \\
1 & 2 & 0 & 0 & 0.3 & 0.7 \\
2 & 1 & 0.6 & 0.4 & 0 & 0 \\
2 & 2 & 0 & 0 & 0.3 & 0.7
\end{bmatrix}
\]

If higher-order effects are known not to be significant, the Markovian property provided an acceptable simplification of the process and makes the process characteristics computationally tractable. The steady-state distribution and collection of mean first passage times for the 2-gram chain are identical to those of the 1-gram chain when you sum over the terminal state.
CHAPTER 4. A HIGHER-ORDER MARKOV PROPERTY

states in the 2-gram. However, if higher-order effects are present, additional considerations
are required. Consider the augmented 2-gram Markov chain below:

\[
P = \begin{pmatrix}
    1 & 1 & 0.5 & 0.5 & 0 & 0 \\
    1 & 2 & 0 & 0 & 0.5 & 0.5 \\
    2 & 1 & 0.5 & 0.5 & 0 & 0 \\
    2 & 2 & 0 & 0 & 0.5 & 0.5
\end{pmatrix}
\]

The changes in the transition probabilities would reflect the fact that the transition from
one state to another does not only depend on the terminal state in the 2-gram, but in the
entire state. Computing the steady-state distribution for the above Markov chain results in
the following:

\[
\Pi = (0.25, 0.25, 0.25, 0.25)
\]

However, when only considering the two atomic states from the original chain, we sum
across the 2-gram states that terminate in the desired state \( i = (1, 2) \). Doing so results in
the following:

\[
\bar{\Pi} = (0.5, 0.5)
\]

4.6 Data-Centric Computational Example

Of interest from a practical perspective is how available data can be used to formulate
a Markov Chain for use in analysis and decision-making purposes. The first major step
involved in developing a Markov Chain is creating the probabilities that will populate the
chain. One possible method for doing so is to examine a time series and draw inference
based on the data. For example, when considering a simple weather example in which
the possible states are \( S = \text{SUNNY}, C = \text{CLOUDY}, R = \text{RAINY} \), we may have the
following time series:

CHAPTER 4. A HIGHER-ORDER MARKOV PROPERTY

If we were considering the single-step transition matrix where the states was simply the current weather, the Markov Chain would provide some description of the weather tomorrow given the weather today. Thus, we are looking to populate the following matrix:

\[
P = \begin{bmatrix}
S & C & R \\
\begin{bmatrix}
p_{SS} & p_{SC} & p_{SR} \\
p_{CS} & p_{CC} & p_{CR} \\
p_{RS} & p_{RC} & p_{RR}
\end{bmatrix}
\end{bmatrix}
\]

(4.12)

where \(p_{ij}\) is the probability of going from state \(i\) to state \(j\) in one step. Using the information in the available time series, we compute \(p_{ij}\) as the percentage of time that state \(i\) is followed by state \(j\). Define \(C(\bullet)\) as the number of times that sequence \((\bullet)\) occurs in a given time series. Note that if state \(i\) is the final data point in the time series, rows containing \(p_{ij}\) will need to be normalized, as no additional sequences beginning with state \(i\) can occur. However, this will only occur for 1 row of any given Markov Chain, corresponding with the terminal data point in the time series.

\[
p_{ij} = \frac{C(ij)}{C(i)}
\]

(4.13)

Using the data in 4.11 we populate 4.12 as follows:

\[
P = \begin{bmatrix}
S & C & R \\
\begin{bmatrix}
1/4 & 1/4 & 1/2 \\
4/5 & 1/5 & 0 \\
1/6 & 1/2 & 1/3
\end{bmatrix}
\end{bmatrix}
\]

(4.14)

Using the steady state formulation given by Resnick [79], the long-term distribution of the weather is as follows:

\[
\Pi = (1, \ldots, 1) * (I - P + ONE)^{-1}
\]

\[
= (1, 1, 1) * \left(\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} - \begin{bmatrix}
1/4 & 1/4 & 1/2 \\
4/5 & 1/5 & 0 \\
1/6 & 1/2 & 1/3
\end{bmatrix} + \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix}\right)^{-1}
\]

\[
= (0.3951, 0.3086, 0.2963)
\]
CHAPTER 4. A HIGHER-ORDER MARKOV PROPERTY

Note that simply counting the number of days in each state (either sunny, cloudy, or rainy) and dividing by the total number of days will yield a different distribution of weather. The distribution is as follows:

\[(S, C, R) = \left( \frac{8}{20}, \frac{6}{20}, \frac{6}{20} \right)\]

\[= (0.4, 0.3, 0.3)\] (4.16)

The differences in the values can be attributed to the Markov Chain taking into account state transitions rather than simply computing event frequency. Intuitively, if we extend the "level" of transition included in the analysis (increasing the amount of information contained in a state description), the values will continue to shift as they account for more information being included in the state description. Construction the Markov Chain in which the state description consists of the previous 2 days yields the following:

\[
P = \begin{bmatrix}
1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/2 \\
1/4 & 0 & 3/4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

(4.17)

Note that the Markov Chain possesses one less state that would be expected. This due to the fact that the event \((CLOUDY, RAINY)\) did not occur in our dataset, so the probability computation associated with that state is equivalent to zero. To make the Markov Chain valid, each row must sum to 1, so a row/column pair consisting of all zeros is consequently deleted. Though care must be taken to append that information to the final state distribution. In order to make an equivalent comparison between \([4.14]\) and \([4.17]\) we compute the steady state distribution of \([4.17]\) and sum across the marginal probabilities of those states that

54
CHAPTER 4. A HIGHER-ORDER MARKOV PROPERTY

terminate in either \((\text{SUNNY}, \text{CLOUDY}, \text{RAINY})\). With the deleted row/column pair re-inserted, the steady-state distribution is as follows:

\[
\Pi = \begin{pmatrix}
0.1111 \\
0.0556 \\
0.2222 \\
0.2222 \\
0.0556 \\
0.1667 \\
0.1111
\end{pmatrix}
\]

(4.18)

Summing across the states terminating in each of \((\text{SUNNY}, \text{CLOUDY}, \text{RAINY})\) yields the following:

\[
\Pi' = (0.3889, 0.2778, 0.3333)
\]

(4.19)

When compared, the steady-state distributions for both Markov Chains 4.14 and 4.17 differ by a non-negligible amount, which reflects the inclusion of additional time series information in the Markov Chain. When the definition of a state in a Markov Chain is expanded to include multiple indexed time periods (referred to in this dissertation as \(n\)-grams), we have shown in the previous section that the computed steady-state distribution varies from the classical single state. We now shift to impacts on the mean first passage times which can be computed from a Markov Chain with multiple time components to each state. The classical formulation of the mean first passage time \(\mu_{ij}\) from some starting state \(i\) to another state \(j \neq i\) as follows:

\[
\mu_{ij} = 1 + \sum_{k \neq j} p_{ik} \mu_{kj}
\]

(4.20)

Note that the computation of the mean first passage time from state \(i\) to state \(j\) is simply the expected value of \(f_{ij}^{(n)} \forall n\). Algebraically,

\[
\mu_{ij} = \sum_{n=1}^{\infty} n f_{ij}^{(n)}
\]

(4.21)
provided that \( \sum_{n=1}^{\infty} n j_{ij}^{(n)} = 1 \). Otherwise, the mean first passage between states \( i \) and \( j \) is infinite. Though this relationship will be discussed in other sections of this dissertation, of interest here is how the relative complexity of the Markov Chain state definitions will impact the mean first passage times of all states \( (i, j) \) in the Markov Chain. The following matrix formulation can be used to compute the collection of mean first passage times for a given transition probability matrix:

1. For an \( n \times n \) transition probability matrix \( P \) over a state space \( s \), the vector of mean first passage times \( [\mu_{ij}] \forall (i, j) \in s \), set the \( j \)th column of \( P = 0 \). Denote this matrix \( \tilde{P} \).

2. Define \( C \) as the cardinality of the state space \( s \). We define \([-1]\) as an \( C \times 1 \) column vector with all entries equal to -1.

3. The solution is computed as follows:

\[
[\mu_{ij}] = (\tilde{P} - I)^{-1} [-1]
\]  

(4.22)

This formulation follows from the standard formulation of the mean first passage times for a discrete-time Markov Chain and makes the following investigation computationally tractable.

### 4.7 Data Sizing to Enable Effective Computation

When the definition of a state in a Markov Chain is extended to include \( l \) multiple time epochs, we initially make the assumption that \( l << n \) where \( n \) is the length of the time series from which the transition probability matrix is computed.

- Need to compute the likelihood of finding any given pattern

- This will give some sense as to how much data is required to observe most pattern at least once.

By enforcing the standard Markovian property, the transition probability matrix can be constructed as follows:
CHAPTER 4. A HIGHER-ORDER MARKOV PROPERTY

1. For each state \( i \in m \), locate each occurrence in the time series \( T \).

2. At every epoch \( t \) where \( i \) occurs, find the state occurring at time \( t + 1 \). This state will by definition be included in \( m \).

3. Define \( O_i \) as the number of times that state \( i \) occurs in \( T \). Also define \( O_{ij} \) as the event of observing state \( j \) immediately following state \( i \).

\[
p_{ij} = \frac{O_{ij}}{O_i}
\]  

(4.23)

Repeating this process for each state pair \( (i, j) \in m \) will yield the transition probability matrix \( P \). We now expand the state definition to include additional information. Rather than defining the state as a single element of \( m \), we define the state now as a pair of elements \( (i_1, i_2) \in m \). However, the transition will still only take one time epoch.

- We informally refer to this as a 2-component state, and we will show that the possible steady state computations are different when considering this state definition versus the standard definition.

Extending the preceding algorithm to an \( l \)-component state allows to include any level of historical information required at the cost of an explosion in the number of possible states. Since the 1-component states are drawn from some collection \( m \) of size \( C(m) \) according to some distribution \( F(m) \). \( F(m) \) can be computed directly from the time series, that is, the probability of encountering any state \( i \in m \) is

\[
p_i = \frac{O_i}{n}
\]  

(4.24)

Consequently, the probability of finding any sequence of states in the time series \( T \) is equal to the product of the unconditional probabilities of each of the individual states. Let us label some sequence of states of length \( l \) drawn from \( m \) as \( S_l \).

\[
P(S_l) = \prod_{i \in S_l} p_i
\]  

(4.25)

I need to here compute how long a time series must be in order to find a given sequence at least \( n \) time in \( T \). This will help provide a bound on how large \( l \) can be when considering information inclusion in the Markov Chain. One critical question to understand is how
CHAPTER 4. A HIGHER-ORDER MARKOV PROPERTY

much information is required to provide a robust understanding of the transition probabilities. Considering \( l \)-component states when the available time series is short will result in a number of states (sequences of length \( l \)) that never occur in the time series. The transition probability matrix resulting from this will be largely populated with zeros. One potential metric for determining the minimum time series length is to compute the likelihood of a sequence \( S_l \). Inverting this probability provides an estimate of the minimum number of observations required to observe the sequence. Performing this for all possible sequences \( S_l \in m \) allows us to choose the minimum length required to observe every sequence. The method for computing this is shown below:

\[
P(S_l) = \prod_{i \in S_l} p_i
\]  

(4.26)

\[
N = \sup_{S_l \in m} \left( \frac{1}{P(S_l)} \right)
\]  

(4.27)

- Once the value of \( N \) is computed, we can recursively examine time series \( T \) and see if it meets the length requirement to consider states of length \( l \).

- For a given time series, we can compute the minimum length required for each state length \( l \).

One additional discussion that bears consideration is how the state space is partitioned. We have up to this point assumed that state space \( m \) is constructed of integers from \( 1, \ldots, i_{\text{max}} \). However, if the time series is shorter than the minimum required length, one possible option is reducing the size of the state space. As a computational example I utilized a state space \( m = (1, 2, 3, 4, 5) \) along with an initial time series length of 100 to compute a lower bound on the time series lengths required to populate a Markov Chain transition probability matrix. One more element that bears discussion is the use of the initial time series \( T \). Using \( T \) initializes the probability vector so as to determine the likelihood of finding a given sequence. This includes the states that are less likely, so the least likely sequence of length \( l \) that can be generated from the states in \( m \) will be constraining factor on how long the time series must be to expect to observe each state. One further line of investigation in this study is how much the required time series length can be reduced if you are willing to give up the likelihood of observing highly unlikely states. At the end of this experiment, an
Figure 4.1: Relationship between amount of data required and the length of the *n*-gram

Figure 4.2: Logarithm of the above graph, confirming the exponential relationship
CHAPTER 4. A HIGHER-ORDER MARKOV PROPERTY

analyst is left with two options if they seek to develop a Markov Chain with $l$-component states:

1. Increase the length of the time series through additional data collection. This is potentially costly, but may be necessary if all states must be considered.

2. Allow highly unlikely states to simply not be expected in the time series. This will result in zeros being found in the Markov Chain transition probability matrix.

4.8 Closing Remarks

This chapter investigated how the application of the standard Markov Property has impacted the development of stochastic processes. Here I also outlined how the property can be extended into a higher-order version. While this concept has been previously utilized in the fields of natural language processing and biology, this chapter marks the first application to business problems. One challenge to applying a higher-order Markov property to business is the amount of data required to generate robust estimates of transition probabilities. The number of required data points grows exponentially as the length of the state description grows. Another challenge when considering states with lengths greater than 1 is how to estimate a form of the mean first passage time between the states. When states are described via n-grams, the transition from one state to another is not the most intuitive result from the problem. Rather, we look to estimate the estimated time from one atomic state to another. This is a more difficult problem, and an area for future research.
Chapter 5

Markov Chain Deflection

5.1 Product Network Interdiction Concepts

Once a network has been created for a product competition as described in this chapter, a possible recourse for a competitor is the removal of a node in the network. One possible reason for removing a network node would be to enact a change in the competitive landscape which would force competitors to respond or provide information to a supply chain or peripheral markets. One recent example of an interdiction is the release of the Apple iPhone 7. For the first time, the iPhone is being released without a built-in 3.5 mm headphone jack. A number of reasons have been offered by Apple as well as by technology columnists. In a forbes.com article dated September 16, 2016 by JV Chamary [16], Apple Chief Marketing Officer Phil Schiller stated:

"Maintaining an ancient, single-purpose, analogue, big connector doesn’t make much sense because that space is at a premium."

By that quote, Mr. Schiller hints at a number of reasons that may underlie the design choice:

1. 3.5mm audio technology gained widespread electronic use in 1964, making the connector more than 50 years old [70].

2. The 3.5mm connector only transmits analog audio signals from a device to a speaker. It doesn’t traditionally serve any other function.
CHAPTER 5. MARKOV CHAIN DEFLECTION

3. The reference to analog technology refers to the modern audio formats being predominantly digital. The 3.5mm audio jack alters the signal before it reaches the speaker, potentially losing information in the transformation [16].

4. Lastly, the 3.5mm connector requires its own power amplifier and digital audio converter, which have recently been built into headphones. Thus, removing the connector makes room for other things [16].

Apple evidently has a vision for the future of mobile and audio technology, and the firm felt the time was right to make this transition [47].

5.2 Literature Review

This chapter presents a method for computing return on investment in a highly stochastic environment. However, approaching the problem requires background research from a number of fields. Research in the evolution of market share dynamics allows us to select a market share model that is mathematically compatible with the investment model to be presented in the next section. Kosobuds and Stokes [52] present a Markov Chain model for oil markets and apply the model to the regulation of cartel behavior across OPEC nations. This work echoes some of the method used in this paper, but we apply the mathematics to computing proactive investment rather than detecting deviation incentives. Aboudi and Thon [1] present an algorithm for detecting dominant and dominating strategies in financial markets. Of application to this paper is the concept of stochastic dominance, which will be used to restrict the strategic investment space in our model. Amable and Verspagen [4] present an empirical analysis attempting to explain the dynamics of market shares through technology innovation. Laursen and Meliciansi [55, 56] add to the previous research by incorporating what they term ”Technological Linkages” or ”Technological Spillovers” into their study of market share dynamics. This concept bears consideration in this paper, as the diffusion of technology throughout an industry will have an impact on the effectiveness of any competitive investment made by a firm. Kato and Hanjo [48] tie market share dynamics to industry concentration, and utilize panel data analysis to explain how highly concentrated industries are less susceptible to rapid changes in market shares. Nguyen and
Shi [68] examined market dynamics using the Bass Diffusion Model as applied to advertising strategies. They considered both market size and market share impacts when attempting to compute the optimal advertising policies. While this paper does not explicitly consider advertising as a motivating factor, the concept of customer shaping is a possible extension to this research.

5.3 Mathematical Model

We present a mathematical model in which the potential market for a product is represented as a discrete-time, discrete state space Markov chain. This allows us to estimate the probability of certain events occurring with regard to the market. Of interest to us is how quickly a market space grows or shrinks from one size \( i \) to another size \( j \). This paper combines those dynamics with a Markovian model of market share, represented by a second Markov Chain.

The underlying assumption of a Markov Chain is the Markovian Property, which for a given stochastic process \( X_t \):

\[
P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \ldots, X_0 = x_0) = P(X_t = x_t | X_{t-1} = x_{t-1}) 
\]

(5.1)

This assumption is fundamental in the results presented above, as it enables the computation of markov chain properties such as mean first passage and absorption probabilities. While it is not possible to prove that this assumption holds for the model, we offer the following argument from experience in both large municipal or government contracts and small consumer goods: The amount allocated to a given good or service depends primarily what was allocated during the preceding time period, rather than in considering multiple time periods in the past. This is true in military and government spending, as well as personal finance.
CHAPTER 5. MARKOV CHAIN DEFLECTION

5.4 Market Growth

The first and simplest version of the problem is that in which the possible states of the market are predefined with a minimum, maximum, and increment value. This allows us to define a finite state space. It also considerably simplifies the computation of Markov Chain properties such as first passage and absorption probabilities. If we represent the market sizes as discrete states \( s \in S \), standard Markov Chain theory provides the probability that a chain will move from state \( i \) to state \( j \) in \( n \) steps as follows:

\[
 f_{ij}^{(n)} = \sum_{k \neq j} p_{ik} f_{kj}^{(n-1)} \forall k \in S
\]  

(5.2)

Additionally, the mean first passage time \( \mu_{ij} \) from some starting state \( i \) to another state \( j \neq i \) as follows:

\[
 \mu_{ij} = 1 + \sum_{k \neq j} p_{ik} \mu_{kj}
\]  

(5.3)

Note that the computation of the mean first passage time from state \( i \) to state \( j \) is simple the extended value of \( f_{ij}^{(n)} \forall n \). Algebraically,

\[
 \mu_{ij} = \sum_{n=1}^{\infty} n f_{ij}^{(n)}
\]  

(5.4)

provided that \( \sum_{n=1}^{\infty} n f_{ij}^{(n)} = 1 \). Otherwise, the mean first passage between states \( i \) and \( j \) is infinite. We can exploit this relationship to begin to answer the question of where a market is likely to grow or shrink to.

5.4.1 Market Share Impacts

Another aspect of this problem to be considered is how investment by a firm will impact the share of the overall market space (considered in the previous section) that is controlled by the firm.

5.4.1.1 Investment Impact on Markov Chain Computations

Under a Markov chain formulation, we can compute the initial and probabilistic long-term market shares for a firm under consideration. However, any investment applied to
CHAPTER 5. MARKOV CHAIN DEFLECTION

The system will impact the market share dynamics. This occurs through increased product attractiveness or novelty as a result of the investment. We model the impact as follows: consider a discrete-time and discrete-state Markov Chain $P$ with state space $(s_1, s_2, \ldots, s_N) \in S$. Each row possesses a distribution describing the probabilistic state transition from time $t$ to time $t+1$, noting that $\sum_j p_{ij} = 1 \forall i$. In this section we propose a model for perturbing each row distribution as a function of investment.

This method involves computing the discrete probability moments for each row distribution. We then shift each of the moments according to some mapping $Z$, and utilize a system of simultaneous equations to recover a new distribution for use in the Markov Chain. It should be noted that this approach represents a discrete version of both the Hausdorff and Stieltjes moment problems. We define the moment generating function as follows:

$$M_X(t) := E\left(e^{tX}\right) = 1 + tE(X) + \frac{t^2E(X^2)}{2!} + \frac{t^3E(X^3)}{3!} + \cdots$$

(5.5)

Computing a given moment $M_n$ using the series expansion above is accomplished by taking the $n$th derivative of the series and evaluating for $t = 0$. This is given by the following formula:

$$M_n = E(X^n) = M_X^{(n)}(0) = \frac{d^nM_X}{dt^n}(0)$$

(5.6)

Theoretically an infinite number of moments can be computed, but restricting the transition probabilities in the Markov Chain to discrete distributions allows us to solve the system using a finite number of moments. The number of moments required is equal to the number of discrete probability partitions in each row distribution of the Markov Chain. After computing the finite number of moments as defined by the number of probability partitions, collect these values into a column vector $C$. For a given level of investment $i$, we define the quantity $\bar{M}$ as the vector of moments after the shift has been applied. Intuitively, the shift in moments due to $M$ can be bi-directional, moving the probability mass in each row towards lower or higher expected values of market share. The goal of Markov Chain interdiction is to shift the probability mass in a direction favorable to the interdicting firm. If the result
of the interdiction shows a negative motion in one or more of the moments, a number of interesting conclusions can be surmised:

- There are market forces operating on the process that cannot be easily quantified.
- The negative shift in one of the moments indicates that the level of investment or interdiction was insufficient to overcome the market forces guiding the process.

We will demonstrate several computational examples in this chapter, but first we require a new mathematical tool to compute the change in moments. In computing the shift in probability associated with a given level of investment \( i \), we borrow from Mechanical Engineering and use the concept of material deformation. In general, we will consider a Markov chain with \( n \) discrete states, restricting the problem to discrete time as well. The transition matrix for the Markov chain is modeled as follows:

\[
P = \begin{bmatrix}
p_{11} & p_{12} & \cdots & p_{1n} \\
p_{21} & p_{22} & \cdots & p_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{n1} & p_{n2} & \cdots & p_{nn}
\end{bmatrix}
\]  

Each row \( i \in (1, \ldots, n) \) of the Markov Chain represents a transition distribution from a state \( i \) to a state \( j \). Given that the sum of each row is equal to 1, we can compute the discrete generalized moments for each row distribution using the formula above. We will use the generalized moments to construct a matrix that we are going to manipulate. We here take advantage of the fact that when constructing generalized moments centered at 0, only the \( n^{th} \) term will be non-zero for the \( n^{th} \) moment. The \( n^{th} \) term of \( M_x(t) \) still has a number of terms corresponding to the number of discrete probability partitions, in this case representing the number of states of the Markov Chain. The expected value of the states of the Markov Chain is also equivalent to the first generalized moment:

\[
E(S) = \sum_{i=1}^{n} p_is_i
\]  

66
CHAPTER 5. MARKOV CHAIN DEFLECTION

For all states $s \in S$, where $p_i$ is the probability of being in state $i$. We will now use this relationship to construct the set of simultaneous equations to be solved in computing a new distribution after investment. The state matrix is as follows:

$$S = \begin{bmatrix} s_1^1 & s_2^1 & \ldots & s_n^1 \\ s_1^2 & s_2^2 & \ldots & s_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ s_1^{n-1} & s_2^{n-1} & \ldots & s_n^{n-1} \\ 1 & 1 & \ldots & 1 \end{bmatrix} \quad (5.9)$$

where $s_i^j$ is the representative value of state $i$ (in this case, market share percentage) raised to the $j_{th}$ power. Note that a unique state matrix can be constructed for each row distribution in $P$. We are now ready to construct the full set of simultaneous equations:

$$S \times P = \begin{bmatrix} M \\ 1 \end{bmatrix} \quad (5.10)$$

$$\begin{bmatrix} s_1^1 & s_2^1 & \ldots & s_n^1 \\ s_1^2 & s_2^2 & \ldots & s_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ s_1^{n-1} & s_2^{n-1} & \ldots & s_n^{n-1} \\ 1 & 1 & \ldots & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ 1 \end{bmatrix} \quad (5.11)$$

Where $M_j$ is the $j_{th}$ generalized moment of the row distribution being considered in $S$. The last row comprised of 1s is necessary in the event that a state is the probabilistic representation of 0 (0% market share, etc.), allowing the set of simultaneous equations to solve for that state as well. Solving the set of simultaneous equations via matrix inversion is straightforward if the matrix entries are unrestricted in sign. However, since the matrix entries here represent discrete probabilities, they must be between 0 and 1. Consequently, there is a limit to how far we can apply a ”deflection” to the set of moments and still retrieve a sensible answer from the matrix inversion.

The set of points in which a feasible solution can be computed is defined as a function of the number of states in the Markov Chain. The number of dimensions needed to describe
the feasible region is 1 less than the number of possible states of the chain. Figure 5.1 shows the feasible region for the solutions when 3 states are possible. In the example given in this paper, these states correspond to (0\%, 50\%, 100\%) of the possible market share for an item. While instructive, this number of states is likely too small to be useful in practical settings. Consequently, the feasible region must be understood for dimensions higher than 2. Using MATLAB software we computed the possible points in $\mathbb{R}^3$ in which \ref{eq:feasible-region} has a solution with $0 \leq p \leq 1 \forall p$. The region in three dimensions is linear and can be defined by the following planes:

5.4.1.2 Moment Deflections

We now turn to the mathematical impact of applying investment in the mathematical model. Standard mechanical engineering methods show the deflection of a cantilever beam to be defined by the equation
CHAPTER 5. MARKOV CHAIN DEFLECTION

\[ y(x) = \frac{Px^2(3L-x)}{6EI} \]  \hspace{1cm} (5.12)

Where \( L \) is equal to the length of the cantilever beam being considered and \( P \) is the force applied at a distance \( x \) from the cantilever point. For this problem we assume a "beam" of length 1, with a point load applied at the end of the beam. Changes in the value of \( P \) are represented by different investment levels applied to the market space. The critical unknown quantity is the value of \( EI \), for which we substitute a constant \( k \). \( k \) is intuitively understood as the "stiffness" of the market space. Elastic markets that are susceptible to changes would possess low values of \( k \), while firmly established markets are likely to have high values of \( k \). In response, one critical computation provided by this analysis is how much investment is likely to affect positive change in the market position of a firm.

5.4.2 Linear Programming Solution

The previous section describes how the application of funding has the potential to change the probabilities in the transition probability matrix of a Markov chain. However, there is a limit to how much the application of funding can change the probabilistic behavior of the chain. This is due to the fact the discrete moments of a distribution can only be deflected a certain amount before a distribution is no longer recoverable. Consequently, we define a constrained optimization problem to estimate how far a Markov chain can be modified. This can be practically interpreted as the point at which investment (in advertising, for example) provides diminishing return. We consider the case where investment applied to the Markov Chain is represented by a scalar variable \( A \). Additional forms of the problem can be constructed when the investment can be represented differently. The construction of the linear program follows from the process of recovering a distribution from a set of moments. Assume that you possess a set of \( n-1 \) moments from a Markov Chain with \( n \) possible states. This is the right-hand side of 5.10. We can recover a set of probabilities by re-arranging the terms of 5.10 as follows:

\[ P = S^{-1}M \]  \hspace{1cm} (5.13)
CHAPTER 5. MARKOV CHAIN DEFLECTION

The process of matrix multiplication in 5.13 allows us to construct the constraints for the linear program. Define \( s_i \forall i \in n \) as row \( i \) of \( S^{-1} \) and \( M = [m_1, m_2, \ldots, m_{n-1}, 1]' \). Since each probability must be \( 0 \leq p_i \leq 1 \) and is computed by multiplying a row of \( S^{-1} \) and \( M \), we construct the constraint as follows:

\[
0 \leq s_i \cdot M \leq 1 \tag{5.14}
\]

Given that we compute the discrete moments \((m_1, m_2, \ldots, m_{n-1})\) for each row of the Markov Chain, the specific values in 5.14 depend on the starting set of moments computed from the Markov Chain. Define \( \bar{M} = (\bar{m}_1, \bar{m}_2, \ldots, \bar{m}_{n-1}) \) as the initially computed discrete moments of row \( \bar{s}_i \). Also define \( A \) as the amount of ”shift” applied to the discrete moments, and \( k = (k_1, k_2, \ldots, k_{n-1}) \) as the vector of stiffness coefficients associated with the market space. Consequently, we write an augmented version of 5.14 as follows:

\[
0 \leq \bar{s}_i \cdot (\bar{M} + \frac{A}{3k}) \leq 1 \tag{5.15}
\]

Since the values of \( \bar{M} \) potentially change for each row \( s_i \in S \), there are \( 2 \times n \) total linear constraints in the problem formulation. Since the only decision variable is \( A \), rearranging the constant and variable terms results in the following two constraints for each 5.15:

\[
\begin{align*}
\frac{S^{-1}}{3k} A & \geq -S^{-1}\bar{M} \\
\frac{S^{-1}}{3k} A & \leq 1 - S^{-1}\bar{M}
\end{align*} \tag{5.16}
\]

With the complete set of constraints given in 5.16, we can now write the complete linear program. The formulation is given as follows:

\[
\begin{align*}
\text{max } & \quad A \\
\text{s.t. } & \quad \frac{S^{-1}}{3k} A \geq -S^{-1}\bar{M} \\
& \quad \frac{S^{-1}}{3k} A \leq 1 - S^{-1}\bar{M}
\end{align*} \tag{5.17}
\]

Matrix \( S \), probability vector \( P \), and moment vector \( M \) are computed for each row in Markov Chain \( P \). Consequently, for \( P \) possessing \( n \) states, we generate \( n \) distinct linear programs based on the computation of discrete moments and shifts based on decision
variable $A$. Given that each row can possess unique sets of moments $M$, the solution to 5.17 in terms of $A$ may differ for each row. Define $A_i$ as the solution to the linear program for row $i \in n$. In order to maintain constraint satisfaction in 5.16 for all rows, we choose the "shift" for the Markov Chain $P$ to be $A^* = \min [A_1, A_2, \ldots, A_n]$. Knowing the value of $A^*$ allows us to re-compute the new set of probabilities in the "shifted" Markov Chain as follows. Define $p_i \forall n \in P$ as row $i$ of Markov Chain $P$.

$$p_i = S^{-1} \left( \bar{M}_i + \frac{A^*}{3k} \right)$$ (5.18)

Computing each row $p_i$ yields a new Markov Chain that has been shifted to the maximum possible value $A^*$.

Here I need to insert the results from the computational experiment, showing how the Markov Chain changes with the different values of $k$.

The steady-state is simple to compute for these chains, one thing I should also add is the mean first passage time computation. Find the other MATLAB script that I wrote to compute mean first passage for the Markov Property Investigation chapter.

### 5.4.3 Long-Run Dynamics

Of interest to firms is the long-run behavior of the Markov Chains, most importantly the stationary distributions possessed by the Markov Chain. In classic stochastic processes this computed as the number of time periods $n \to \infty$, but in many applications firms will not be able to wait until the system reaches steady state. Alternatively, we consider the dynamics of how a Markov Chain converges over time to the steady state, reaching an approximate steady-state condition in a smaller number of steps. The textbook by Resnick [79], among other sources, provides a straightforward algorithm to compute the stationary distribution $\Pi$ of the Markov chain. Let $P$ be the $m \times m$ irreducible stochastic matrix representing the Markov chain. Include a row vector comprised of exactly $m$ 1’s. Let $I$ be the $m \times m$ identity matrix, and let ONE be the $m \times m$ matrix consisting entirely of 1’s.

$$\Pi = (1, \ldots, 1) \ast (I - P + \text{ONE})^{-1}$$ (5.19)

This algorithm assumes that the quantity $(I - P + \text{ONE})$ possesses an inverse, the verification of which is provided in [79] and thus omitted here.
5.4.4 Computational Example

Consider the following Markov Chain, defining the step-wise transition between discrete levels of market share. In this case the states are defined as \((s_1, s_2, s_3) := (0, 0.5, 1)\). The initial transition probability matrix is as follows:

\[
P = \begin{bmatrix}
0.5 & 0.25 & 0.25 \\
0.25 & 0.5 & 0.25 \\
0.25 & 0.25 & 0.5
\end{bmatrix}
\]  

(5.20)

We also define a vector of values \(k\) as follows:

\[
k = (2, 4, 8)
\]  

(5.21)

The standard long-term behavior computations for the above matrix, without deflection, are as follows:

\[
\Pi = \begin{pmatrix}
1 & 1 & 1 \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{pmatrix}
\]

\[
M = \begin{bmatrix}
3 & 4 & 4 \\
4 & 3 & 4 \\
4 & 4 & 3
\end{bmatrix}
\]

When we apply our deflection scheme to the matrix via the a linear programming formulation, we return a different Markov Chain.

\[
P = \begin{bmatrix}
0.4355 & 0.3145 & 0.25 \\
0.1855 & 0.5645 & 0.25 \\
0.1855 & 0.3145 & 0.5
\end{bmatrix}
\]

This yields a different Markov Chain than before, as an external force has been applied.

\[
\Pi = (0.2473, 0.4194, 0.3333)
\]

\[
M = \begin{bmatrix}
4.0435 & 3.1795 & 4 \\
5.3913 & 2.3846 & 4 \\
5.3913 & 3.1795 & 3
\end{bmatrix}
\]
CHAPTER 5. MARKOV CHAIN DEFLECTION

This resulted from a maximum deflection of $A = 0.1935$

Intuitively, if the Markov chain in 5.20 defines the time-dependent market share associated with some product, then investment in that product (in the form of research and development, etc.) will change the entries in the transition probability matrix. One major question remaining is how the vector of $k$ values is computed for each scenario. The ”flexibility” of a given market space is context dependent, and it bears research to discern a robust method of computing the values of $k$. In this computational example we compute $k$ as a series of values which increase exponentially. The value of $k$ is in the denominator in the deflection equation, so increases in the values of $k$ represent increased ”stiffness” as the moment number increases. This can be described as follows:

$$k = k^\alpha \forall i \in S$$ \hspace{1cm} (5.22)

Increases in the value of $\alpha$ indicate that the market space is getting ”stiffer” as the moment number increases. This formulation assumes that the values of $k$ in the vector are all correlated in logarithmic space, which in practicality may not be true.

5.4.5 Stiffness Vector Computation

When utilizing this deflection problem formulation in an applied context, it is critical to incorporate time series data. An appropriate example for this dissertation is the fluctuation in the stock price of GoDaddy.com (from finance.yahoo.com). GoDaddy gained some notoriety for their advertising expenditures during the Super Bowl of each year. This makes the time series of stock prices an good example of periodic deflections, which in this case are representative of advertising. Given that we know a deflection occurs around the date of the Super Bowl each year, we here attempt to compute the stiffness of the market space. For each year period preceding and following the advertising deflection in 2016, we
construct the one-step transition probability matrices (denoted $P_1$ and $P_2$).

$P_1 = \begin{bmatrix}
0.75 & 0.25 & 0 & 0 & 0 \\
0.33 & 0 & 0.66 & 0 & 0 \\
0 & 0.03 & 0.87 & 0.1 & 0 \\
0 & 0.01 & 0.07 & 0.87 & 0.05 \\
0 & 0 & 0 & 0.92 & 0.08
\end{bmatrix}$

$P_2 = \begin{bmatrix}
0.75 & 0.25 & 0 & 0 & 0 \\
0.33 & 0 & 0.66 & 0 & 0 \\
0 & 0.03 & 0.87 & 0.1 & 0 \\
0 & 0.01 & 0.07 & 0.87 & 0.05 \\
0 & 0 & 0 & 0.92 & 0.08
\end{bmatrix}$

For the 2016 Super Bowl, the average amount spent on advertising per company was estimated at $10 million. Using this as a first-order estimate of the "force" applied to the process, we can back-calculate the values in the vector $k$ that affected the transition from
CHAPTER 5. MARKOV CHAIN DEFLECTION

$P_1$ to $P_2$. The results are as follows:

$$k = \begin{bmatrix}
-18,222,222 & -357,298 & -9,330 & -273.79 & -8.56 \\
-1,439,393 & -23,709 & -525.43 & -13.19 & -0.36 \\
24,393,939 & 328,839 & 6,120 & 131.3 & 3.06 \\
-3,042,168 & -47,504 & -985.43 & -22.91 & -0.57 \\
-46,620 & -1,303 & -36.42 & -1.02 & -0.03
\end{bmatrix}$$

The above matrix of $k$ values bears some discussion. A negative entry implies the existence of a market force that the advertising investment was not able to overcome. Entries moving rightward indicate higher-order moments, representative of how variance, skewness, and kurtosis of the process move. The example shown above only considers 1 interdiction and should not be considered a robust estimation of market stiffness or the forces therein. This example primarily serves as a proof-of-concept for the idea of stochastic process interdiction.

5.5 Discussion

The computation of return on investment is critical in many business applications. The key shortcoming that this research addresses is the investment in a firm changes the competitive environment as time passes. Previous methods of quantifying investment return have not considered this phenomenon. One area for future research when utilizing this methodology is the practical significance of a moment deflection reaching a boundary of the feasible region.
Chapter 6

Network Formulation of Competition

6.1 Competitive Network Introduction

This chapter addresses some shortcomings in modern competitive contracting. First, the tools and techniques used by proposal developers are often qualitative. The analyses developed and their strategic implications depend largely on the judgment of professional subject matter experts (SMEs), who are intrinsically subject to bias and error. This research seeks to remediate that by providing quantitative tools with which to understand complex competitions. Second, the integration of network science and network interdiction methods allow proposal developers to understand the global impacts of individual requirement or competition changes, far beyond what is currently possible. Tools and techniques from the field of graph theory have the potential to provide an analytical framework by which large-scale competitions and strategies are analyzed and optimized.

6.1.1 Problem Motivation

The field of Competitive Strategy has undergone a significant transformation in past decades to adapt to an increasingly interconnected global economy. Business development professionals and executives re-evaluate business techniques in order to remain competitive. Alongside changes in the business climate, academic researchers have contributed conceptual and algorithmic developments to how companies strategize about retaining cur-
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

rent or winning future business. A survey of the concept is provided by Bolton and Ockenfels [1], in which they note the shortcomings and possible research directions of applying auction and matching theory to complex firm decisions. One component of competitive strategy has been referred to as customer shaping. This term refers to an effort by a competitor to influence the priorities of the customer. The end goal of a competitor is to place himself or herself in a competitively advantageous position. A majority of the academic research on the topic of customer shaping is qualitative. We hypothesize that this is the case for two major reasons: (1) a firm’s strategies with regard to customer shaping are likely competition sensitive and firms are remiss to release the pertinent numerical data, and (2) most published strategies to date acknowledge the complexity of understanding the needs of a customer under incomplete information. It is important to realize that not all competitions possess the complexity to allow or require strategic specifications manipulation as it is considered in this paper. If a competitive scenario is complex enough to allow requirements definition and manipulation, which can be the result of a complex product and/or a complex contract structure, the methods developed in this paper can be brought to bear to improve a competitor’s strategic position. Engelbrecht-Wiggans [2] provides a survey of the taxonomy of auction and bidding models available. Page 124 of the article mentions that Sequences of auctions with more than one player are much more difficult [than those with one player]; There are essentially no results in this area [2]. The first application of solving network interdiction problems was in the military domain [1, 2]. As battlefields grew in size, military commanders realized the increasing need to protect an entire network of supplies and personnel. As networks have grown in recent decades in both scale and pervasiveness, a number of approaches to network interdiction problems exist in the literature. We classify the approaches into two general categories: solutions in which arcs are removed and those in which nodes are removed. Wood [3] presented some of the first results on an interdiction problem. Israeli and Wood [4] present algorithms to solve the removal of arcs from a network in order to maximally affect the length of the shortest path. The analysis of node removals has generated a larger volume of literature. Borgatti [5] uses social networks as the environment in which to investigate the impact of removing nodes from a network. In doing so, he identifies key players in a social network, whose removal maximally impacts the graph-theoretic properties of the network. The detection of critical
nodes in a graph is also approached by Arulselvan, et al. [6], who identify the theoretical complexity constraints on the problem as well as present a heuristic for solving the problem in reasonable time. Di Summa, et al. [7] present an integer programming model to attempt to maximize the disconnection between graph nodes, as well as theoretical results regarding the model. Dinh, et al. [8] present an algorithm which allows for the combined removal of nodes/arcs. A common thread in the network interdiction methods is the fact that the algorithms lie beyond P-SPACE, not even being bounded by a constant in integrality [9]. This spurred the development of alternative methods and transformation, as well as heuristics [10]. Although fundamentally distinct from network interdiction, network control is also growing as a field of research. Network scientists are beginning to understand principles guiding complex networks. Liu and Barabasi [11] lay out the theoretical foundations for complex network control, though their focus is on networks that tend to self-organize based on size and survival and growth processes. The current paper provides an additional method to deal with the situation when the rules of the game change mid-game. This renders a number of bidding/auction approaches obsolete, as they rely on static game structure. We approach the problem of auctions and bidding from a highly computational perspective, acknowledging the difficulty of developing an elegant, closed-form solution.

6.1.2 Game Theoretic Analysis

While there are many permutations to defining a competitive environment, the model definition must be general enough to cover many possible instantiations and specific enough to permit computational analysis. For each competition, we consider a product that divided into a number of sub-regions that represent characteristics or requirements of the product as a whole. As an example, consider a hammer: designing a hammer means a selection of materials for head, shaft, and grip (if applicable) materials. Upon a walk through the local home improvement store, not all combinations of materials (for example, a wooden shaft with a rubber grip) are common or even feasible, and an even larger number of those combinations are cost-prohibitive. Consequently, the graph representing the competition will not be a complete graph. However, what is required for a feasible solution is a collection of choices that include each of the sub-regions defined by the product. The figure below
shows two possible solutions for an abstract product: Each sub-region will contain \( n \geq 1 \) nodes, and arcs between nodes constitute inclusion in a feasible solution. At present, we will restrict the model to only permit 1 node in each sub-region for each feasible solution.

The graph will have arcs connecting nodes between different sub-regions, and intra-sub-region arcs are not permitted. A development provided by this paper is the fact that the arcs in the graph are matrix-weighted, i.e. there is a game-theoretic matrix associated with each arc representing the payoffs earned by each of 2 players by selecting that arc in a feasible solution. Consequently, the feasible solutions selected by each player can be distinct, depending on the payoffs in the arc matrices. In the following section, we discuss notation and formal model construction.

### 6.2 Mathematical Model Development

In this chapter, we consider requirements and performance specifications defined over an undirected graph. The undirected graph shown below will serve as the foundation for the model development and analysis. With regard to notation, the matrices accompanying each arc denote the payoffs in a 2-player game. The matrix is as follows:

\[
\begin{bmatrix}
x_{1,1} & x_{1,2} \\
x_{2,1} & x_{2,2}
\end{bmatrix}
\]  

(6.1)

where \( x_{(i, j)} \) equals the game payoff where players 1 and 2 choose the strategy combination \((i, j)\). In this example each player is assumed to have binary strategy options (whether or not to traverse each arc in the network). Following the arc payoff reduction algorithm given in Harris and Kamarthi [12], each node is subsequently reduced to a scalar value.
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

representing the expected payoff of the game. The dashed lines represent sub-regions of the graph. We write graph algebraically as follows: $G = (V, E)$ where $V$ is the collection of vertices (nodes) and $E$ is the collection of edges (arcs). The sub-regions of the graph will be denoted $M$, and it should be noted that

$$\bigcup_i M_i (V_i, E_i) = G (V, E)$$

The objective of this problem is to choose a feasible collection of nodes $F$ such that $F$ contains exactly one node from each sub-region. The graph used as an example in this paper is as follows: After node payoff reduction, the graph becomes the following: Note that the reduction methodology allows for both pure and mixed strategy profiles. Based on the definition of payoffs in a two-player, zero-sum game, player 1 will consequently seek the set of nodes that maximizes the sum of node values in a tree. Conversely, player 2 will seek the minimum-sum set of nodes from the reduced graph. This problem does not equate to the classical minimum spanning tree, as not all feasible nodes in the graph must be included in a feasible solution.
6.2.1 Heuristic Development

A large percentage of network interdiction problems possess NP-hard complexity [3]. Thus, the availability of heuristic is important in the event that an instantiation of the problem lies beyond current computational limits. One intuitive heuristic is to reduce each of the regions to contain only the node that contains the greatest (for player 1) or least (for player 2) node weight, defined as the sum of all the weights of the arcs incident on the node. This reduces the problem to the computation of a minimum (or maximum) spanning tree, but at the cost of losing connectivity information regarding the network. Consequently, the computation of a heuristic solution will be rapid, but is not guaranteed to be optimal. This is shown next.

6.2.2 Complete Algorithm Development

1. Assume a connected, positive-weighted graph $G(V, E)$ with vertex set $V$ and edge set $E$.

2. Split $G$ into a set of sub-regions, each denoted as $G_i \subset G$, each of which contain a subset of nodes $V_i \subset V$ and a subset of edges $E_i \subset E$ such that $\bigcup_i G_i = G$.
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

3. Let \( e_{jk} \) be the arc length between nodes \( j \) and \( k \). Within each region \( G_i \subset G \), temporarily connect all pairs of nodes \((j, k)\) residing in \( G_i \) and assign \( e_{ij} = 0 \forall j, k \in G_i \).

4. Select an arbitrary node as a starting point, and use Prims algorithm to compute the minimum spanning tree of \( G \). Let \( T \) be the minimum spanning tree of the graph \( G \), and let \( T(v) \) and \( T(e) \) be the vertices and edges in \( T \).

5. For all nodes \( v \in T(v) \), compute the incident weight \( \sum_{e,v} \) for each node, which is the sum of the weights of all incoming arcs. If \( \sum_{e,v} = 0 \), remove the node and adjacent arcs from the graph.

6. Check if the number of nodes in each sub-region \( G_i = 1 \). If so, the algorithm has returned an optimal solution and terminates. Otherwise, go to step 7.

7. For a sub-region \( G_i \) with more than one node, the removal of any node must result in a graph disconnection (proposition 1, proven below). For each node \( v \in G_i \), compute the sub-region of \( G \) that will disconnect from with the removal of \( v \).

8. Re-label to the newly disconnection subgraph as \( H \). Consequently, determine which of the \( n \) nodes in \( G \) have the minimum length by which to re-connect \( G \) and \( H \). Reconnect \( G \) and \( H \) using the arc of minimum length that spans both subgraphs.


6.2.3 Proof of Optimality

Proving that the algorithm above returns the optimal solution required 2 steps: Proposition 1: For a sub-region \( G_i \) containing \( n > 1 \) nodes, the removal of any node from \( G_i \) will disconnect a sub-region from \( G \). Proof: This proof follows from the construction of the minimum spanning tree of \( G \) and the general properties of the minimum spanning tree. Steps 4 and 5 of the algorithm generate a graph composed of a minimum spanning tree with potentially a number of nodes and arcs removed. Removing a node and its incident arcs will only fail to disconnect a graph in the presence of a cycle. By definition, a minimum
spanning tree does not contain a cycle. Therefore, the only remaining the question to consider is whether removing nodes with incident weight $\sum_e v = 0$ could add a cycle to the graph. Since step 5 only removes arcs, it is impossible for the operation to introduce a cycle to the graph. Therefore, removing any node from $G_i$ will disconnect the graph. Proposition 2: $T^*$ is the optimal solution to the sub-region graph problem (i.e. no shorter set of arcs exists which create a connected graph having one and only one node in each sub-region $G_i$.) Proof: This proof follows from the fact that the algorithm above reduces regions with more than one node to a single node by removing nodes from V with non-minimum incident weights. By definition, a minimum spanning tree is the set of arcs which connects all the nodes in a graph using the minimum total edge weight. The algorithm above departs from the computation of a minimum spanning tree in two places, which we must consider in order to prove $T^*$ as the optimal solution. The algorithm examines sub-regions with $n > 1$ nodes contained in them and reduces the number of nodes by 1 with each iteration. Proposition 1 shows that each node reduction creates a disconnected graph. Consequently, the reconnection of the disconnected subgraphs, each of which do not contain cycles, will not create a cycle in the new graph. The algorithm then re-introduces the shortest candidate arc to reconnect $G$ and $H$. Therefore, the algorithm reduces the graph to a standard tree without sub-regions, of which the minimum spanning tree is the collection of arcs which connects all nodes with the minimum arc weight.

6.3 Sensitivity Analysis

Assuming an analyst is able to construct payoff matrices as shown in Figure 2a, it is natural to inquire about the sensitivity of the payoffs. In many circumstances, the payoffs will have resulted from qualitative estimates of the outcome of an interaction between the two players. The previous discussion about the solving for the minimum and maximum feasible solutions assumes a current set of payoff matrices given a priori. This assumption may be reasonable at a given time epoch in a competition, but it is of great importance to understand the sensitivity of optimal solutions as the competitive landscape changes. Understanding the impact of matrix value perturbation is critical in practical applications, as
we often do not know the precise values of the matrices. Considering the graph from Figure 2a, we will examine the impact of node perturbation on the optimal minimum solution. By applying the algorithm given above, we can arrive at the minimum solution of nodes 1, 7, 3 with an overall length = 11.5. Looking back at the arcs that connect this feasible solution, we examine the question of what are acceptable ranges for each of the matrix values so that the identified solution remains optimal. When considering only scalar values, the process of sensitivity analysis is relatively straightforward, simply varying the scalar value. However, matrices add considerable complexity. We propose three methods for investigating sensitivity in these circumstances:

1. variation in individual matrix entries
2. variations in particular rows/columns
3. variation of the entire matrix by a constant factor

The algebraic procedure for each is similar.

6.3.1 Individual Value Sensitivity

Considering first the variation of a single matrix value, we will need to resolve the game-theoretic strategy problem again, but with an additional variable added. Initially we must solve for the critical value that will cause the current optimal solution to become sub-optimal. This requires finding the minimum solution as well as the second-smallest solution to the problem. For example, the graph in Figure 2b has a minimum solution of 1, 6, 3 with length = 11.5, and second-smallest solution of 7, 2, 5 with length = 19. We will consider the required change in the upper-left entry in the matrix associated with the arc connecting nodes 1, 6. A $2 \times 2$ game-theoretic matrix will possess either a pure or a mixed strategy equilibrium point. For example, consider the matrix linking nodes 1 and 6 from Figure 2b, which possesses a mixed strategy equilibrium point at $(x_1, x_2) = \left( \frac{5}{9}, \frac{4}{9} \right)$ and $(y_1, y_2) = \left( \frac{1}{3}, \frac{2}{3} \right)$ which yields an arc value of

$$\left( \frac{5}{9} \right) \left( \frac{1}{3} \right) 7 + \left( \frac{5}{9} \right) \left( \frac{2}{3} \right) 3 + \left( \frac{4}{9} \right) \left( \frac{1}{3} \right) 1 + \left( \frac{4}{9} \right) \left( \frac{2}{3} \right) 6 = \frac{13}{3} \quad (6.3)$$
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

If we need the arc value associated with this matrix to increase and exceed 11.8, we will investigate by how much an individual value in the matrix will have to increase to make this possible. Increasing the arc value beyond 11.8 would cause solution 1, 6, 3 to become sub-optimal and cause solution 7, 2, 5 to become optimal. Considering the matrix below, we make a number of observations about the impacts of perturbing the values in the matrix below:

\[
\begin{bmatrix}
a & b \\
c & d
\end{bmatrix}
\]  \hspace{1cm} (6.4)

Since the matrix either possesses a mixed or pure equilibrium point, perturbing any individual value will either (1) have no impact on the type of equilibria found, or (2) cause an optimal pure strategy profile to become mixed or vice versa. Considering the second case, we demonstrate that a perturbation of a matrix value causes a change in the equilibrium type and we only need to be concerned about the critical values at which this process takes place. Considering the matrix connecting nodes 1 and 6 from Figure 2a, and varying the value of \(b\) (in this case, \(b = 3\)) yields the following: Pure strategy equilibrium points are in bold. Because the matrix above starts with a mixed strategy equilibrium point, increasing the value of \(b\) will eventually cause strategy \(x_1\) to dominate strategy \(x_2\). This occurs when \(b = 6\), and indicates that player 1 sees no net gain by choosing a strategy other than \(x_1\). Increasing \(b\) to, say, \(b = 6.5\) would cause player 1 to select \(x_1\) and player 2 to select \(y_2\). However, note that \(b\) cannot increase indefinitely, as player 2 would simply shift their choice to strategy \(y_1\) when \(b\) exceeds 7. After that, value \(b\) can increase without bound, and player 2 still finds it optimal to remain with strategy \(y_1\). For the general arc matrix below, we define the following:

\[
\Gamma = \begin{bmatrix}
p_{11} & p_{12} \\
p_{21} & p_{22}
\end{bmatrix}
\]  \hspace{1cm} (6.5)

\[
\hat{p} = \max \{p_{11}, p_{12}, p_{21}, p_{22}\}
\]  \hspace{1cm} (6.6)

\(\Delta e_{ij}\) = the value by which the expected payoff for the matrix would have to increase (or decrease, in the case of minimization) for the current network solution to become sub-optimal. Proposition 2: If \(\hat{p} < \Delta e_{ij}\), the perturbation of a single value cannot augment the matrix and in turn cause the current network solution to become sub-optimal. Proof: In
cases of both pure and mixed strategy equilibrium, we compute the expected payoff from
the matrix by
\[
E(\Gamma) = \sum_{i=1}^{2} \sum_{j=1}^{2} x_i y_j p_{ij}
\]
(6.7)
where \(\{X = (x_1, x_2), Y = (y_1, y_2)\}\) are the optimal mixed strategy profiles for players 1
and 2, and \(p_{ij} = \Gamma_{ij}\). \((x_i, y_j)\) are defined \(\forall i, j \in [0, 1]\). Consequently, the maximum
value that \(E(\Gamma)\) can take occurs when \((x_i, y_j) = (1, 1)\) and \(\Gamma(i, j) = \hat{p}\). This is equal to
\(E(\Gamma) = (1)(1)\hat{p} = \hat{p}\). As stated, \(\hat{p} < \Delta e_{ij} \forall p_{ij} \in \Gamma\). □

### 6.3.2 Row and Column Sensitivity

In the case when entire rows or columns of the matrix are perturbed, the above is not
necessarily true. When augmenting a single value, an opposing player can simply alter their
strategy and limit the payoff impact. However, when augmenting entire rows or columns,
this benefit disappears, and the augmented matrix can potentially change the optimality of
the current network solution. Similar to the perturbation of a single value, changing the
values in a row or a column by some value, say \(k\), will eventually drive a matrix from
a mixed-strategy solution (if it currently exists) to pure strategy equilibrium. Using one
of the matrices from Figure 2a provides some insight into the mechanics of network and
matrix perturbation by rows and/or columns. Consider the general matrix below, as well as
the more general form to the right:
\[
\begin{bmatrix}
  a_1 & a_2 \\
  a_3 & a_4
\end{bmatrix}
\]
(6.8)
Assuming the matrix possesses a mixed-strategy equilibrium point, we define the following:
\[
r_1 := [a_1, a_2], \quad \Delta(r_1) := |a_1 - a_2|
\]
(6.9)
\[
r_2 := [a_3, a_4], \quad \Delta(r_2) := |a_3 - a_4|
\]
(6.10)
\[
c_1 := [a_1, a_3], \quad \Delta(c_1) := |a_1 - a_3|
\]
(6.11)
\[
c_2 := [a_2, a_4], \quad \Delta(c_2) := |a_2 - a_4|
\]
(6.12)
Accordingly, the following definitions are helpful in computing the changes necessary to move from a mixed-strategy to a pure-strategy equilibrium point. $F(X)^+ = $ the critical change that will cause an equilibrium type shift (mixed to pure) in the positive direction. $F(X)^- = $ is the equivalent change in the negative direction. We define the critical values as follows:

$$F(r_1)^+ = \min [\Delta(c_1), \Delta(c_2)]$$  \hfill (6.13)

$$F(r_1)^- = -\max [\Delta(c_1), \Delta(c_2)]$$  \hfill (6.14)

$$F(c_1)^+ = \max [\Delta(r_1), \Delta(r_2)]$$  \hfill (6.15)

$$F(c_1)^- = -\min [\Delta(r_1), \Delta(r_2)]$$  \hfill (6.16)

By observation, the following symmetric relationships are true:

$$F(r_2)^+ = -F(r_1)^-$$  \hfill (6.17)

$$F(r_2)^- = -F(r_1)^+$$  \hfill (6.18)

$$F(c_2)^+ = -F(c_1)^-$$  \hfill (6.19)

$$F(c_2)^- = -F(c_1)^+$$  \hfill (6.20)

Consequently, computing one critical value will immediately yield another. We now compute the critical values for the matrix connecting nodes 1 and 6 in Figure 2a (below).

$$\begin{bmatrix} 7 & 3 \\ 1 & 6 \end{bmatrix}$$

$$\Delta(r_1) = |7 - 3| = 4$$

$$\Delta(r_2) = |1 - 6| = 5$$

$$\Delta(c_1) = |7 - 1| = 6$$

$$\Delta(c_2) = |3 - 6| = 3$$
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

<table>
<thead>
<tr>
<th>Row or Column number</th>
<th>Maximum Increase</th>
<th>Maximum Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row 1</td>
<td>3</td>
<td>-6</td>
</tr>
<tr>
<td>Row 2</td>
<td>6</td>
<td>-3</td>
</tr>
<tr>
<td>Column 1</td>
<td>5</td>
<td>-4</td>
</tr>
<tr>
<td>Column 2</td>
<td>4</td>
<td>-5</td>
</tr>
</tbody>
</table>

Table 6.1: Computed Perturbation Values for the matrix connecting nodes 1 and 6 in 6.2

Using the formulae defined in 6.13-6.20:

\[
F(r_1)^+ = \min [\Delta(c_1), \Delta(c_2)] = \min [6, 3] = 3
\]
\[
F(r_1)^- = -\max [\Delta(c_1), \Delta(c_2)] = -\max [6, 3] = -6
\]
\[
F(c_1)^+ = \max [\Delta(r_1), \Delta(r_2)] = \max [4, 5] = 5
\]
\[
F(c_1)^- = -\min [\Delta(r_1), \Delta(r_2)] = -\min [4, 5] = -4
\]

By the symmetry relations noted above, we compute the remaining four perturbation relationships as shown below:

\[
F(r_2)^+ = -F(r_1)^- = 6
\]
\[
F(r_2)^- = -F(r_1)^+ = -3
\]
\[
F(c_2)^+ = -F(c_1)^- = 4
\]
\[
F(c_2)^- = -F(c_1)^+ = -5
\]

Summarizing the results above, we have now computed the ranges in which each row and column of the strategy can be perturbed without changing the equilibrium type (moving from mixed to pure strategies).

The computations above are helpful for understanding the equilibrium points and decision profiles of the two players in the game, but we need an additional step to understand when a change in the optimal network solution will occur. Two additional definitions are helpful for computation. We define the value of the game as follows:

\[
G(F(\phi_i)^*) \text{ for } \phi = \{c, r\}, i = \{1, 2\}, * = \{+, -\}
\]

Intuitively, \( G \) is equal to the game value at equilibrium when any of the above perturbations are applied. Define \( T_1 \) as the minimum network tree, \( T_2 \) as the second smallest network...
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

tree, \(e_{ij}\) as the arc associated with the matrix being perturbed, and \(\Delta \Gamma\) as the required change in the game-theoretic matrix value to make \(T_1\) sub-optimal.

\[
\Delta \Gamma = F(\phi_i)^* + ((l(T_2) - l(T_1 \setminus e_{ij})) - G(F(\phi_i)^*))
\]  

As noted earlier in this section, the optimal minimum solution to the graph in Figure 2 includes nodes \(\{1, 6, 3\}\) and has a length = 11.5. Additionally, the second-shortest solution includes nodes \(\{7, 2, 5\}\) and has a length = 19. In an effort to change the optimal solution, we will augment the matrix connecting nodes 1 and 6. The current game-theoretic value of the matrix is 4.3, and for the solution to become sub-optimal, the value of the matrix must be greater than \((19 - 7.2) = 11.8\). By proposition 2, the matrix value cannot exceed the maximum individual entry value (in this case, 10). Consequently, only a row or column perturbation will cause the value to increase to 11.8 and beyond. Because we can compute the above relationship for \(\phi = \{c, r\}, i = \{1, 2\}\), all four values are necessary to completely understand matrix sensitivity. We are not considering the case where the matrix rows and columns are changed in a negative direction \(\ast = \{-\}\), as that would make the length of the minimum tree increasingly small. We compute the values for above-mentioned matrix as shown below:

\[
r_1^+ : \Delta \Gamma = F(r_1)^* + ((l(T_2) - l(T_1 \setminus e_{ij})) - G(F(r_1)^*))
\]

\[
= 3 + (11.8 - 6)
\]

\[
= 8.8
\]

Therefore, row 1 of the matrix would have to be increased by 8.8 in order to make the current network solution sub-optimal; by incorporating all rational changes in strategy that would accompany changes in payoffs. Next, we compute the other three possible perturbation values for the matrix \(\{r_2, c_1, c_2\}\), in the same form:

\[
c_2^+ : \Delta \Gamma = 4 + (11.8 - 7) = 8.8
\]

\[
r_2^+ : \Delta \Gamma = 6 + (11.8 - 7) = 10.8
\]

\[
c_1^+ : \Delta \Gamma = 5 + (11.8 - 6) = 10.8
\]

There is symmetry in the values: when the pairwise perturbations to \(\{r_1, r_2, c_1, c_2\}\) have the same value, the rows and columns intersect in the off-diagonal entries, in this case, \(\{c_1, r_2\}\).
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

and \((r_1, c_2)\). Observation will verify that when decreasing rows or columns in the matrix (possibly when solving for maximum tree solutions), the intersection of rows and columns with the same value will occur on the diagonal of the matrix. Row and column perturbation may reflect a lack of understanding about specific strategic impacts or interdependencies and consequently is of interest to an analyst. Scaling an entire matrix by some constant \(k\) is also possible, but is of less strategic interest because a perturbation of this type will eventually change the optimal network solution, but will never change the strategic profile of the matrix. This is because the relationship between the individual matrix entries never changes when we scale the matrix by \(k\).

6.3.3 Additional Sensitivity Considerations

While all of these events may occur in accordance with some stochastic process, another pertinent change mechanism is an intentional change to the graph by a rational player. The literature refers to this as network interdiction. The formulation of the interdiction problem in this case relies on the integer programming formulation of the minimum spanning tree problem known as Subtour Elimination. Intuitively this integer programming formulation exploits the property of the minimum spanning tree containing \((N - 1)\) arcs, where \(N\) is the number of nodes in the graph. The method seeks to eliminate all simple cycles from the graph by enumerating all node combinations examining each to assure the total number of arcs selected is equal to \((N - 1)\). As expressed by Golari [56], the Integer Program is as follows:

\[
\begin{align*}
\min_x & \quad \sum_{(i,j) \in E} \phi_{ij} x_{ij} \\
\text{s.t.} & \quad \sum_{(i,j) \in E} x_{ij} = N - 1 \\
& \quad \sum_{(i,j) \in E(S)} x_{ij} \leq |S| - 1, \forall S \subset V, S \neq V, S \neq \emptyset \\
& \quad x_{ij} \in \{0, 1\} \forall (i, j) \in E
\end{align*}
\] (6.23)

where \(E(S) \subset E\) is a subset of edges with both ends in subset \(S \subset V\). The second constraint ensures that there are no simple cycles for all combinations of nodes in the graph. This program will return the collection of arcs that constitute the minimum spanning tree of
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

graph $G(V, E)$. However, computing the optimal interdiction profile (the optimal subset of nodes to delete in order to maximize the length of the remaining minimum spanning tree) requires a modification to the above model. The following algorithm will heuristically accomplish the task:

1. For an undirected graph $G = (V, E)$ with vertices $V$ and edges $E$, we will remove $n$ nodes from the graph.

2. Sort all edges $e_{ij} \in E$ in ascending order of their length and store them in list $A$.

3. Since a minimum spanning tree will have exactly $|V| - 1 - N$ edges, select the shortest $|V| - 1 - N$ edges in another list $B$ and consider them in the graph. This will potentially create a disconnected graph.

4. For the arcs in $A \setminus B$, find the first arc that, if added, will connect disconnected components of the graph. Repeat step 4 until all components are connected.

Note that the above heuristic might not return the minimum spanning tree of the graph, as adding arcs to connect disconnected components will increase the length of the tree returned by the algorithm. Another potential weakness of the heuristic is the addition of cycles in the graph. The addition of a single node between disconnected components will not create a simple cycle, but repeating step 4 multiple times could create a cycle. A cycle would indicate a non-minimum spanning tree of the graph, as the length of the cycle would be greater than $(N - 1)$. The development of the optimal interdiction program is still an area of open research and is beyond the scope of this dissertation.

6.4 Computational Example: MBTA Operations

One possible application for such an algorithm is a decision analysis toolkit for competitions in which a product must meet a number of design requirements. We attribute the regions in the graphs discussed above to different requirements that a product must possess in order to be a feasible design. Intuitively, the set of decisions needed to generate a feasible solution to the problem will entail a number of interdependencies, represented by the
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

<table>
<thead>
<tr>
<th>Factor Number</th>
<th>Factor from MBTA Document</th>
<th>Alternative 1</th>
<th>Alternative 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transportation Services</td>
<td>TR₁</td>
<td>TR₂</td>
</tr>
<tr>
<td>2</td>
<td>Materials Management and Procurement</td>
<td>MM₁</td>
<td>MM₂</td>
</tr>
<tr>
<td>3</td>
<td>Quality</td>
<td>Q₁</td>
<td>Q₂</td>
</tr>
<tr>
<td>4</td>
<td>Operator Customer Service</td>
<td>OCS₁</td>
<td>OCS₂</td>
</tr>
<tr>
<td>5</td>
<td>Management and Personnel</td>
<td>MP₁</td>
<td>MP₂</td>
</tr>
<tr>
<td>6</td>
<td>Training Services</td>
<td>TS₁</td>
<td>TS₂</td>
</tr>
<tr>
<td>7</td>
<td>Construction Support</td>
<td>CS₁</td>
<td>CS₂</td>
</tr>
<tr>
<td>8</td>
<td>Mobilization</td>
<td>MOB₁</td>
<td>MOB₂</td>
</tr>
<tr>
<td>9</td>
<td>IT Requirements</td>
<td>IT₁</td>
<td>IT₂</td>
</tr>
</tbody>
</table>

Table 6.2: Keolis vs. MBCR Operating Criteria, alternatives representing node labels

In 2013, the state of Massachusetts opened competition for the operations contract to run the Massachusetts Commuter Rail System. Two major corporations met the initial requirements and entered into competition for the contract, valued at more than $280 million per year (USD) [95]. Keolis and MBCR entered into competition submitting a statement of qualification and responding to a formal request for proposal issued by the Massachusetts Bay Transit Authority (MBTA). On January 8, 2014, the board of the MBTA awarded a $2.68 Billion contract to Keolis Commuter Services, with the possibility for two 2-year extensions that could bring the total contract value to $4.3 Billion [76].

This paper will explore the question of whether the competition could have ended differently had MBCR modeled the customer requirements using a graph formulation. All documents related to the competition are in the public domain and available online at www.mbta.com. The requirements for such a system are myriad and complex, so we only consider a requirement subset here. When considering only the evaluated factors from Schedule 3 of the operating agreement, the graph contains 16 regions. A number of these regions are likely to have a single required strategy from each competitor (e.g. Safety and Security, Service Level Agreements), so they are not included in the analysis. For brevity and without loss of generality, we limit the number of regions here to nine. The list of factors assigned to sub-regions is as follows: Defining specific strategic plans for each al-
alternative in each category is beyond the scope of this paper, but that process would take place in practical competition. Note that most factors included in the above table have an impact on the total evaluated price against which the customer will evaluate the overall proposals. Consequently, the weights for the nodes in the total evaluated price sub-region will have high values. The graph of the sub-regions shown in 6.4 contains a number of interdependencies as well. For example, a prospective competitor’s choice of a Management and Personnel strategy will likely affect their transportation services and operator customer service regions as well. Subject Matter Experts (SME) define the weights for the matrices, but for this instance, the weights will reflect the competitive impacts of incumbency and performance. We then apply the algorithm developed earlier to the matrix to discern the optimal strategies for each player. The interdependencies shown on the graph above demonstrate how complex these analyses can become, even for a simple example. As an initial computational analysis, a MATLAB implementation of the algorithm presented in section 4 was developed and tested on the undirected graph in Figure 8. The analysis incorporated randomness into the process by generating random integer off-diagonal values, and biasing the diagonal values towards the competitor perceived to have an advantage between the two strategy choices. For example, if Keolis (player 2) possesses a competitive advantage over MBCR (player 1) when considering the link between personnel and training, the
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

![Arc Frequency among 500 replications of the Model](image)

Figure 6.5: Frequency of arcs from the Monte Carlo simulation (sorted in the descending order)

The arc weight matrix could look something like the following:

$$
\begin{bmatrix}
0 & 6 \\
5 & -6
\end{bmatrix}
$$

The $-6$ value in the lower right entry reflects the competitive advantage assessed, while the off-diagonal entries (5 and 6) demonstrate the uncertainty allocated to the process. The top left entry is representative of the negligible competitive impact if neither player chose the arc in their competitive strategy. To model the uncertainty in the process, we incorporate Monte Carlo simulation to demonstrate commonality in the solutions. The graph below demonstrates how certain arcs appear more often in a higher number of optimal minimum solutions: The information shown in 6.4 does not add much clarity to the discussion until we couple it with the visual description of the results. Figure 10 visually represents the graph by thickening arcs that appear more often amongst the 500 performed replications of the model. 6.4 shows how often a particular set of arcs tend to appear in optimal solution while finding the minimum distance between each sub-region by including only one node from each. The strong relationship between system mobilization (region 4) and IT requirements (region 5) is visually and mathematicallly intuitive due to the fact that region 5 has only one feasible sub-region connecting to it. It is of interest to note that the choice of nodes and arcs to traverse between regions four and five depended strongly on the other decisions made in traversing other parts of the graph. The table below includes a subset of arc frequencies and confirms the results shown above.
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

Figure 6.6: Solution frequency using Monte Carlo simulation

<table>
<thead>
<tr>
<th>Node-to-Node</th>
<th>Relative Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-9</td>
<td>0.26</td>
</tr>
<tr>
<td>7-17</td>
<td>0.258</td>
</tr>
<tr>
<td>8-18</td>
<td>0.258</td>
</tr>
<tr>
<td>8-17</td>
<td>0.254</td>
</tr>
<tr>
<td>8-10</td>
<td>0.252</td>
</tr>
<tr>
<td>7-9</td>
<td>0.248</td>
</tr>
<tr>
<td>7-10</td>
<td>0.24</td>
</tr>
<tr>
<td>7-18</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Table 6.3: High frequency arcs and relative appearance values

After the arc connecting nodes 7 and 18, the remaining arcs in the graph appear in the minimum spanning tree much less frequently, reflecting the thesis of this paper that we can detect and analyze critical components of competitive strategy using network science and graph theory principles. After applying the algorithm discussed previously in the paper, the optimal solution for both player 1 (Keolis) and player 2 (MBCR) is the following selection of strategies and their corresponding nodes. Additionally, in the course of this competition, each of the competitors optionally submitted a document suggesting modifications to the Operating Agreement. A customer normally offers this when they attempt to solicit cost

95
savings from prospective bidders, as the bidders have an opportunity to demonstrate how a change in a requirement could result either in a lower Total Evaluated Price, or in additional performance for the same price. Both bidders responded to the offer, though only MBCR suggested a substantive change to the Operating Agreement. Keolis’s choice not to make substantive changes could be representative of confidence in the contract structure and competitive position, while MBCR’s opposing choice could indicate what they perceive as a weakness, namely, an inability to rapidly upgrade the computing and IT infrastructure.

The implementation of Prim’s algorithm in this paper relies heavily on an adjacency matrix representation, which makes the computational complexity $O(|V|^2)$ where $V$ is the number of nodes in the graph $G$. Although the algorithm suggests sequentially moving from sub-region to sub-region and removing sub-optimal nodes, we found it much more computationally efficient to execute the following steps: (1) identify nodes for removal after the initial application of Prim’s algorithm; (2) remove all identified nodes in one step; and (3) reapply Prim’s algorithm to develop/confirm the final solution. This simply adds a factor $k$ to the complexity of the algorithm, depending on the number of times the algorithm executes Prim’s algorithm. The graph in Figure 11 shows the computation time required for
1,000 Monte Carlo iterations of the model, with a total execution time of 16 minutes and 42 seconds.

6.5 Network Games Discussion

This paper demonstrates the power of incorporating graph theory, operations research, and game theory. In complex competitions, these methods are critical for understanding the global impacts of changes to the competitive landscape. This method, however, is not without shortcomings. Practitioners must have the ability to translate competitive intelligence information into the matrices that accompany the arcs in $G(V,E)$. This process is not necessarily straightforward, as estimates on the arc weights are subject to intrinsic bias.

We propose the incorporation of techniques such as the Analytic Hierarchy Process (AHP) and/or Conjoint Analysis to augment the development of the arc matrices presented in this paper. The incorporation of these techniques will improve a contractors understanding of what factors are important to the customer and will consequently shape the competitive strategy developed using the aforementioned network analysis techniques. One consideration that bears on future research is the fact that complex systems, such as those considered in this paper, have a large number of factor permutations. This makes both the Analytic Hierarchy Process and Conjoint Analysis a time-intensive exercise. Mitigation strategies for this include truncating the set of factors considered in a model construction like Figure 7, or using probabilistic inference methods to fill in a large number of values for the AHP or Conjoint Analysis. Both options are open research questions.

Even without the current incorporation of either the Analytic Hierarchy Process or Conjoint Analysis, two factors help to build confidence in the estimates. One is the fact that matrix entries associated with dominated strategies in pure-strategy matrices do not influence the final computed arc weight. In other words, for some game instantiation represented by player choices $(x_i, y_j) = p_{ij}$, if either $x_i$ or $y_j = 0$, than that strategy combination will not occur and the associated payoff value with not factor into computing the expected payoff value associated with proposition 2. The second factor is the fact that an analyst can easily implement Monte Carlo simulation and sensitivity analysis to investigate whether any un-
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

certainty in a matrix will affect the optimal network solution. This chapter accomplishes the following as a unique addition to the literature.

1. The algorithm presented in this paper differs from previous network interdiction algorithms. Unlike in previous methods, the arcs are matrix-weighted, which can accurately reflect multiple entities competing along the same arcs. This stands in contrast to the attacker-defender concept, where the two aforementioned entities try to traverse and deny the network, respectively. In this algorithm, both players can occupy the same node simultaneously.

2. The use of heuristics can result in a rapid solution, but it is easy to find an example where a heuristic will fail to find the optimal solution. Consequently, the analytical solution presented in this paper remedies a heuristic shortcoming and provides a provably optimal solution in polynomial time.

3. Full industrial application of these methods will require a concerted effort from subject matter experts to populate the matrix weights, but this can in turn yield a very full picture of the competitive reality. Additionally, encouraging subject matter experts to consider the pairwise impacts of binary strategic choices is likely to increases the fidelity and practicality of competitive intelligence information. We propose this as an interesting area for future research.

4. In the computational example presented earlier in the paper having to do with the Massachusetts Bay Transportation Authority (MBTA), competitive bidders hardly tried to make changes to the contract structure. One definite area of future research will be a computational investigation of whether the results of the competition in which Keolis was awarded the MBTA operating contract would have been different had one or more nodes been interdicted in the network formulation.

5. I acknowledge that uncertainty in the estimates for the matrix weights is still likely to cause hesitation in implementing the methodology presented in this paper. To address the concerns, we used sensitivity analysis to overcome some of the uncertainty by identifying critical thresholds, which can change the strategic choices defined by a node. Strategic changes have the potential to alter the outcome of the network games.
CHAPTER 6. NETWORK FORMULATION OF COMPETITION

6. One major benefit to utilizing this method is that it can accurately temper optimism or pessimism regarding potential competitive discriminators. When considering a competitive factor, we must understand its comprehensive impact on the competition before quantifying its effects. We can attribute one of the two characteristics to each factor in the network: we should consider a factor in depth (represented in this paper by an arc) because it has the potential to change the optimal network solution, or, regardless of the amount of research and analysis done on the factor, it cannot change the optimal solution.

This dissertation seeks to integrate quantitative methods from operations research and network analysis with qualitative methods from competitive intelligence and game theoretic method in a globally optimal way. In doing so, we have opened up the possibility for firms involved in competitive contracting to bring considerable computational resources to bear on competitive decision-making.
Chapter 7

Conclusion

This dissertation takes a number of views on competitive business scenarios when rational players seek to make a change to the structure of the game.

7.1 Summary of Dissertation Topics

The following section gives a brief summary of the topics presented in this dissertation. Additionally, thoughts on integrating the tools and industrial contexts are provided.

7.1.1 Corporate and Competitive Strategy

The ability of a firm to compete effectively depends greatly on how it understands the entire technical and market landscape. Mathematical tools will provide a number of different avenues with which to understand the data (or lack thereof) and make appropriate and defensible decisions in response. The section in this dissertation presents two optimization models for integrating corporate and competitive strategy. The first model allows a firm to estimate the subset of available capabilities that should included in a portfolio in order to create competitive and cost-effective products. A novel addition to this is the ability to also consider pairwise contributions that are brought by each capability. The second model considers a related decision to the first: how to allocate research and development investment to currently held capabilities such that their competitive contributions are maximized.
CHAPTER 7. CONCLUSION

Both models are presented in a matrix form, which implies that they are easily extensible to large-scale problems.

7.1.2 Markov Property Shortcomings

The definition of a Markov Chain state is critical in determining real-world accuracy. Choosing too simple of a state for the sake of computability will cost precision in process capability. Choosing too complex of a state for the sake of accuracy will require an enormous number of states coupled with additional data to ensure a non-sparse transition probability matrix.

7.1.3 Network Game Formulation

When firms must develop complex products, the interrelated decisions in the designs present considerable complexity when trying to discern to find optimal solutions. Optimality in these scenarios is not only a product that meets the customer requirements, but one that is competitive when compared to the product of another firm. The addition of a network formulation to product design enables the inclusion of game theoretic and graph theory techniques in the analysis. This enables a firm to not only maintain a global view of their product, but effectively estimate their market position.

7.1.4 Markov Chain Interdiction

The concept of a Markov Chain Interdiction invites a firm to be an active participant in the Markovian process of which they are a part. The major contribution to the section in this dissertation is the presentation of a method for estimating the impact of such activity. Because interdiction in most market spaces is difficult to predict, developing techniques to aid long-term choices is a major advantage.
CHAPTER 7. CONCLUSION

7.2 Computational Integration

The computational methods developed in this dissertation enable improved decision making in complex competitions. Additionally, the application of these tools can be combined in a number of scenarios. For example, considering the MBTA computational example from the network games formulation chapter, the acquisition or divestiture of a capability from the network can be modeled as shown in 7.2.

Figure 7.1: Computation integration of 3.3 and the network game formulation analysis. The figure shows how capabilities could be added or removed in order to change the optimal competitive design decisions.

7.3 Industrial Contexts

The process of a contractor responding to a customer request underpins the mathematics in this dissertation, and the tools presented potentially apply differently depending on the process stage. Prior to the award of a contract, a number of well-defined tasks could benefit from this research:

1. Price-to-Win: understanding the complex interactions between design elements will benefit a firm seeking to calculate its competitive position. The ability to concurrently consider the actions and capabilities of a competitor is a major contribution to this research, and will benefit firms as a market space grows more competitive.
CHAPTER 7. CONCLUSION

2. Voice of the Customer: Ensuring alignment with customer requirements will also be aided by the models presented in this dissertation. The mathematical models developed here will allow a firm to consider individual approaches to meeting customer requirements, and result in an improved competitive position through the optimization of the network.

3. Operational Analysis: improving the competitive portfolio or more accurately the impact of interdiction in stochastic processes will add resolution in the modeling and simulation area. The ability to consider much greater levels of system design (i.e. the acquisition or divestiture of entire technologies or capabilities) is a powerful display tool for demonstrating system performance.

The tools presented in this dissertation intuitively apply best when computed prior to the award of a contract. However, they can also be used on historical artifacts to identify areas of competitive strength or weakness and the corresponding business results. Of particular interest is the ability to identify the factors that contributed most to either successful or unsuccessful competitive results.

7.4 Future Research Questions

The sections presented in this dissertation serve as a starting point in analyzing business with imperfect information when the players involved look to interdict in the processes. There are a number of extensions to the research here that bear investigation.

1. The integration of the optimization problems in 3.3 and 3.4 into a single decision framework would be beneficial to a firm seeking to utilize the technique.

2. Extending the analysis of a high-order Markov process also bears investigation. Results in this area have the potential to improve upon the shortcoming of standard Markov chains. Possible applications of a robust higher-order formulation include improved long-term predictions and market analysis concepts.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


[47] KASTRENAKES, J. The iPhone 7 has no headphone jack, 2016.


BIBLIOGRAPHY


BIBLIOGRAPHY


[70] PECKHAM, J. iPhone 7 headphone jack: why did Apple drop it?, 2016.


BIBLIOGRAPHY


