2013

Stochastic network interdiction games

Zheng, Jiefu

Boston University

http://hdl.handle.net/2144/11094

Boston University
STOCHASTIC NETWORK INTERDICTION GAMES

by

JIEFU ZHENG

M.S, Nanjing University, 2007

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2013
Success is stumbling from failure to failure with no loss of enthusiasm.
– Winston S. Churchill
Acknowledgments

There are so many people that I want to thank for their helps, which are essential and necessary for me to complete this dissertation.

First and foremost, I would like to thank my advisor, Professor David Castañón for his guidance and support. He brought me into the fantastic topic of network introduction and provided me a positive environment to complete this research. David has broad and deep knowledge. He can always provide innovative ideas and insightful guidance, which help me to boost the research to a higher level. Under his guidance, I have the confidence to handle any challenge in research.

I would like to thank all committee members. Prof. Vakili, Prof. Caramanis and Prof. Nazer were also the committee members in my prospectus defense. Their questions and suggestion were insightful, which helped the development of this research. Prof. Paschalidis taught me optimization techniques, which are useful tools to show our results theoretically. In addition, I would like to thank Prof. Vakili, whose coordination is vital for me to pursue this degree.

I would like to express my gratitude to all the teachers and instructors in Boston University. I also like to thank the funding from the Air Force Office of Scientific Research (AFOSR), which provided me financial support in the past three years.

Last I sincerely appreciate the support from my family, especially my wife Qi Wang. In these years she undertook lots of housework, including taking care of our daughter Alice. In addition, she always gave me substantial faith before I lost confidence. Without her support, it’s impossible for me to finish this thesis.
STOCHASTIC NETWORK INTERDICATION GAMES

(Order No. )

JIEFU ZHENG

Boston University, College of Engineering, 2013

Major Professor: David A. Castañón, PhD,
Professor of Electrical and Computer Engineering

ABSTRACT

Network interdiction problems consist of games between an attacker and an intelligent network, where the attacker seeks to degrade network operations while the network adapts its operations to counteract the effects of the attacker. This problem has received significant attention in recent years due to its relevance to military problems and network security. When the attacker’s actions achieve uncertain effects, the resulting problems become stochastic network interdiction problems. In this thesis, we develop new algorithms for the solutions of different classes of stochastic network interdiction problems.

We first focus on static network interdiction games where the attacker attacks the network once, which will change the network with certain probability. Then the network will maximize the flow from a given source to its destination. The attacker is seeking a strategy which minimizes the expected maximum flow after the attack. For this problem, we develop a new solution algorithm, based on parsimonious integration of branch and bound techniques with increasingly accurate lower bounds. Our method obtains solutions significantly faster than previous approaches in the literature.

In the second part, we study a multi-stage interdiction problem where the attacker
can attack the network multiple times, and observe the outcomes of its past attacks before selecting a current attack. For this dynamic interdiction game, we use a model-predictive approach based on a lower bound approximation. We develop a new set of performance bounds, which are integrated into a modified branch and bound procedure that extends the single stage approach to multiple stages. We show that our new algorithm is faster than other available methods with simulated experiments.

In the last part, we study the nested information game between an intelligent network and an attacker, where the attacker has partial information about the network state, which refers to the availability of arcs. The attacker does not know the exact state, but has a probability distribution over the possible network states. The attacker makes several attempts to attack the network and observes the flows on the network. These observations will update the attacker’s knowledge of the network and will be used in selecting the next attack actions. The defender can either send flow on that arc if it survived, or refrain from using it in order to deceive the attacker. For these problems, we develop a faster algorithm, which decomposes this game into a sequence of subgames and solves them to get the equilibrium strategy for the original game. Numerical results show that our method can handle large problems which other available methods fail to solve.
Contents

1 Introduction ................................................................. 1
   1.1 Network Interdiction Problems .................................. 1
   1.2 Problems of Interest ............................................. 4
   1.3 Contributions .................................................... 6
   1.4 Dissertation Overview ......................................... 8

2 Background .................................................................. 10
   2.1 Stochastic Network Interdiction ................................ 10
   2.2 L-Shaped Decomposition ........................................ 19
   2.3 Sample Average Approximation ............................... 22
   2.4 Games in Extensive Form ........................................ 24
   2.5 Point-Based Value Iteration .................................... 32

3 Branch and Bound Algorithms for Stochastic Network Interdiction Problems ........................................ 36
   3.1 Problem Formulation .............................................. 37
   3.2 Modified Branch and Bound Method .......................... 39
   3.3 Numerical Results ................................................ 51

4 Extended Stochastic Network Interdiction Problems ........................................ 57
   4.1 Interdiction of Undirected Networks .......................... 57
   4.2 Interdiction of Multi-source/destination Network .......... 61
   4.3 Interdiction of Networks with Uncertain Sources and Destinations ........................................ 67
   4.4 Solutions to the Extended Models ............................... 69
# 5 Multi-Stage Interdiction Problems

5.1 Problem Formulation ........................................... 76
5.2 Model-Predictive Approach .................................. 78
5.3 Branch and Bound for the Approximation Problem ........... 82
5.4 Numerical Results ............................................. 86
5.5 Extended Multi-stage Interdiction Problems .................. 92
  5.5.1 Interdiction of Undirected Networks ................. 93
  5.5.2 Interdiction of Multi-source/destination Networks .... 95
  5.5.3 Interdiction of Networks with Uncertainties Source/destination .... 97
  5.5.4 Solution to the Extended Models ..................... 97

# 6 Dynamic Network Interdiction Games with Nested Information

6.1 Problem Formulation ........................................... 102
6.2 Subgame Decomposition Method ............................... 113
  6.2.1 Subgame Equilibrium .................................. 113
  6.2.2 Identifying the Set of Support Vectors .............. 121
  6.2.3 Retrieving the Global Saddle Point Strategies ....... 130
  6.2.4 Subgame Decomposition on a Two-interaction Game .... 136
6.3 Other Nested Information Games .............................. 139

# 7 Games with Markov structure

7.1 The Common “Cost-to-go” Problems .......................... 152
7.2 The Indifferent Histories ..................................... 157
7.3 Numerical Results .......................................... 160

# 8 Conclusions and Future Work

8.1 Conclusions .................................................. 167
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.2 Future work</td>
<td>169</td>
</tr>
<tr>
<td>Appendices</td>
<td>171</td>
</tr>
<tr>
<td>A The proof of Theorem 6.2.9</td>
<td>172</td>
</tr>
<tr>
<td>References</td>
<td>176</td>
</tr>
<tr>
<td>Curriculum Vitae</td>
<td>179</td>
</tr>
</tbody>
</table>
List of Figures

2·1 The interdiction problem on a directed flow network. The numbers associated with each arc are the capacity of that arc and the probability of removing that arc if it's under attack. The defender will send flows from \( s \) to \( t \) after the attack. ........................................ 11

2·2 The extensive form of a network interdiction game. .......................... 27

3·1 The sequence of actions of the one-stage interdiction problem. ........ 36

3·2 MBB solving an interdiction problem on a network with 5 attackable arcs where the attacker can attack at most 3 arcs. .................. 50

3·3 SNIP 7 \times 5: 37 nodes and 72 arcs, 22 of which are interdictable. .... 52

3·4 SNIP 4 \times 9: 38 nodes and 67 arcs, 24 of which are interdictable. .. 53

4·1 IEEE Bus 300 System: an undirected flow network with 409 arcs and 300 nodes, out of which 33 are sources (dark rectangles) and 36 are sinks (light diamonds). Capacities of arcs are adjusted such that all demands are met before attack. .............................. 74

5·1 The sequence of actions of the two-stage interdiction problem. ........ 77

5·2 Flow reductions \( r \) versus approximation rankings of the top 1% strategies in the approximation problems. ............................. 81

6·1 The sequence of actions of the dynamic network interdiction problem with nested information. .................................................. 105

6·2 Realization plans linearize the realization probability of a play \( \sigma^T \). 106
6.3 Decomposing a multi-stage game into a one-stage subgame and sub-games with one-stage less (represented by rectangles). \((x^1, y^1)\) are first interaction strategies and \((x^r, y^r)\) are remaining interaction strategies. 114

6.4 Bottom up estimation of the set of support vectors \(Q_{1x,t}\) ............... 122

6.5 Estimating the value function \(V(\beta^{1x,t})\) with the support vectors on randomly sampled distributions \(\beta\) ......................... 129

6.6 Top down retrieval of the global equilibrium. ........................................... 131

6.7 Subgame decomposition on an extensive form game. ......................... 137

7.1 Grouping nodes (within the circles) with the same latest state to reduce the sizes of subgames. ................................................................. 157

7.2 The underlying networks for the nested information game. For each arc, the first number is the capacity, and the second number (if any) is the probability of broken if attacked. Arcs without this number are not interdictable. .................................................. 161
**List of Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>Dynamic Programming</td>
</tr>
<tr>
<td>i.i.d</td>
<td>identical independent distribution</td>
</tr>
<tr>
<td>IP</td>
<td>Integer Programming</td>
</tr>
<tr>
<td>LP</td>
<td>Linear Programming</td>
</tr>
<tr>
<td>MBB</td>
<td>Modified Branch and Bound Method</td>
</tr>
<tr>
<td>MIP</td>
<td>Mixed-Integer Program</td>
</tr>
<tr>
<td>PBVI</td>
<td>Point-Based Value Iteration</td>
</tr>
<tr>
<td>POMDP</td>
<td>Partially Observed Markov Decision Process</td>
</tr>
<tr>
<td>SAA</td>
<td>Sequential Approximation Algorithm</td>
</tr>
<tr>
<td>SAM</td>
<td>Sample Average Approximation</td>
</tr>
<tr>
<td>SMIP</td>
<td>Stochastic Mixed-integer Program</td>
</tr>
<tr>
<td>SNIP</td>
<td>Stochastic Network Interdiction Problem</td>
</tr>
<tr>
<td>$x \cdot y$</td>
<td>inner product of the same dimension vectors $x$ and $y$</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Network Interdiction Problems

The problem of network interdiction arises when an attacker, using limited resources, tries to reduce the functionality of an intelligent network (also called the defender) that can observe the attack and adapt its operations to preserve functionality. Such problems have received much attention in recent years due to the interest in many contexts such as cyber security, military logistics, anti-drug operations, power distribution and communication networks. In military deployment, the attacker destroys the enemy’s supply network with physical attacks involving aerial sorties, ground troops or cruise missiles. For communication networks, other mechanisms such as jamming of receivers or software attacks would be used to reduce functionality. In anti-smuggling operations, border police would like to attack the cross-country smuggling network to thwart the illegal drugs transportation. The study of this problem also helps the prevention of the infection within hospitals (Assimakopoulos, 1987).

People have been studying network interdiction problems for decades. The famous max-flow min-cut problem (Ford and Fulkerson, 1962) can be viewed as the simplest version of network interdiction problem where the attacker wants to use least resource to cut off the network flow entirely from a source to a destination. Since then, people have developed many network interdiction models to emphasize different aspects of the problem. These models vary in the objectives of the attacker and the intelligent network as well as the ways that the attack can affect the network.
One class of models is the minimum detection problem, where the attacker can place sensors on arcs to increase the probability of detection while the defender selects the path with minimal detection probability after observing the attacker's action. Washburn and Wood (Washburn and Wood, 1994) formulated this problem as a two-person zero-sum game. They showed that the equilibrium mixed strategies for both players can be found by solving a minimum-cut problem. Pan et al. (Pan et al., 2003) studied an extended model where the source and destination are unknown to the attacker. They reformulated it as a two-stage stochastic mixed-integer program with recourse and showed the problem is strongly NP-Hard. Morton et al. (Morton et al., 2007) developed a stochastic version of this problem to thwart nuclear smuggling where the defender is aware of only a subset of the sensor locations.

Another class of models is the shortest path network problem, where the defender wants to travel on the shortest path between a source and a destination while the attacker wants to make this path as long as possible by interdicting network arcs using limited resources. Fulkerson and Harding (Fulkerson and Harding, 1977) found that when the lengths of arcs can be changed linearly to the interdiction cost, this problem is equivalent to a minimum cost flow problem. Israeli and Wood (Israeli and Wood, 2002) studied the same problem with binary interdiction and extended it by allowing unknown source/destination pairs. Held et al. (Held et al., 2005) extended the model for random networks where the objective is to maximize the probability that the minimum path length from a source to a destination exceeds a given threshold. They implemented a scenario decomposition method to solve this problem. They also described and compared heuristic solution methods for multi-stage versions in (Held and Woodruff, 2005).

In the minimum max-flow problem, the attacker seeks to minimize the maximum flow between source(s)/destination(s) by interdicting arcs in the network. An early
version was studied in (Wollmer, 1964), where the attacker was allowed to remove a fixed number of arcs from the network. Another extension of this model to allow removing a fixed number of nodes was considered in (Corley and Chang, 1974). Wood (Wood, 1993) showed that under a cardinality constraint on the number of arcs that can be interdicted, this problem is NP-Complete in a strong sense.

Cormican et al. (Cormican et al., 1998) extended the problem from deterministic interdiction to stochastic interdiction with binary outcomes after arcs are attacked. In this formulation, attacked arcs are destroyed with known probabilities, and the defender observes the outcomes of the attack, and then sends flow over the network. They developed upper and lower bounds to the value function achieved by the attacker, which were used to construct a sequence of solutions that approach optimal attack performance.

Lim and Smith (Lim and Smith, 2007) studied two multi-commodity problems: one with discrete interdiction and the other with continuous interdiction. They proposed an optimal partitioning algorithm along with a heuristic procedure for estimating the optimal objective function value. Akgün et al. (Akgün et al., 2011) considered a deterministic interdiction problem between a set of nodes. They developed an exact method, which solves a mixed-integer problem converted from the bi-level min-max interdiction problem. They also provided an approximation method that partitions the nodes into disjoint subsets such that each node group is in a different subset and minimizes the sum of the arc capacities crossing between different subsets. The approximation is good in terms of similar post-interdiction flow capacities.

Many of the above stochastic interdiction problems can be formulated as stochastic mixed integer programs. Thus, some randomized algorithms like (Sanchez and Wood, 2006), (Mak et al., 1999), (Verweij et al., 2003), (Rocco and Ramirez-Marquez, 2009) can be used to solve these network interdiction problems.
1.2 Problems of Interest

In many applications, the maximum flow between a source and a destination is more important than whether there is a flow between them. For example, in anti-drug smuggling, the border police want to reduce the quantity of smuggled drugs. In military deployment, we want to decrease enemy's supply as much as possible. Therefore we focus on minimum max-flow interdiction problems.

In this thesis, we are investigating three types of network interdiction problems. The first problem we consider is the stochastic network interdiction problem where an attacker chooses a set of arcs to attack with limited resources, to minimize the maximum flow that the network will be able to support after the attack. The outcome of the attack is random and binary, i.e., each arc being attacked is either removed with a given probability or remains in the network with full capacity. We assume that the outcomes are independent across arcs. This stochastic model is more suitable for problems where the interdiction result cannot be completely controlled by the attacker.

Numerous methods have been developed to solve this model, like the Sequential Approximation Algorithm in (Cormican et al., 1998) and the Sample Average Approximation in (Janjarassuk and Linderoth, 2008). However, owing to their computational complexity, they do not scale well to large problems. Therefore we develop a new solution method that integrates the branch and bound techniques with increasingly accurate lower bounds to find the optimal interdiction strategy. Numerical experiments show that our method is about two orders of magnitude faster than the Sequential Approximation Algorithm. Notice the underlying network in the basic model is directed and has a single source and a single destination. To incorporate different aspects of real applications, we also extend the model to interdiction problems on undirected networks, multi-source/destination networks and networks with
uncertain source/destination.

The second class of problems that we address is the multi-stage network interdiction problem where an attacker can attack multiple times and observe the results of its attacks. This is an extension of the previous stochastic network interdiction problem which allows the attacker to adapt its next attack(s) based on the observed outcomes of previous attacks. There was little work in the literature to emphasize the impact of the updated information on sequential attacks. Cormican et al. (Cormican et al., 1998) studied a multiple attack problem in their extended model, but they did not allow the attacker to observe the outcomes of its previous attacks, and thus did not generate adaptive strategies.

Our problem formulation is similar to the stochastic resource allocation model studied in (Castañón and Wohletz, 2009). We develop a model-predictive approach as in (Castañón and Wohletz, 2009). Instead of solving the exact model, we first solve a lower bound approximation which allows for non-integer attack allocations in the second and subsequent waves, but requires integer allocations in the first stage. The optimal solution to the lower bound problem is a binary strategy for the first attack. After carrying out that strategy and observing the result, the attacker has an interdiction problem with one less stage. By doing this iteratively, one can generate binary strategies for all stages. We focus on the two-stage interaction problem and develop a new set of performance bounds, which are integrated into a branch and bound procedure that extends the single stage approach to multiple stages. We extend the model to incorporate the interdiction problems on undirected networks, multi-source/destination networks, uncertain source/destination networks and develop solutions for these extended models.

In the previous problems, the attacker has full information of the network state. A third class of problems that we address in this thesis is the interdiction problem
where the attacker has only statistical information about the network state, referred as partial information. For instance, the attacker may not observe the result of all attacks, or the attacker is uncertain about the existence of certain arcs. In these problems, the attacker may gather information from observing the actions of the network. This raises the possibility for the network to deny information to the attacker.

This problem can be abstracted as a zero-sum game with nested information. To find the equilibrium behavior strategy, we adopt the Linear Programming (LP) formulation developed in (von Stengel, 1996). Notice the size of the LP problem grows exponentially with the number of interactions. To tackle this difficulty, we develop a method, which exploits the nested information structure of the game and decomposes the multi-stage games into a sequence of one-stage subgames. The method estimates the expected payoff of each subgame as a function of its initial belief. We show that the original equilibrium strategies can be found by solving these subgames. For games with Markov structure, experiments shows that our method is faster than the comparable LP approaches in (von Stengel, 1996). For larger problems which the LP approach fails to solve, our method can output high quality solutions within a short time.

1.3 Contributions

The contributions of this thesis can be summarized as follows:

- For one-stage stochastic network interdiction problems,
  - Developed a new solution method that integrates the branch and bound techniques with increasingly accurate lower bounds, which is shown to be significantly faster than the previous methods.
  - Extended the model to undirected networks, multi-source/destination networks, uncertain source/destination networks and developed new solution
algorithms for these extended models.

- For multi-stage network interdiction problems with adaptive strategies
  - Proposed a new formulation to address the problem where the attacker can adapt its next actions based on the observed outcomes of previous attacks.
  - Developed a model-predictive approach and extend our previous method to solve this problem. Numerical experiments show that our method is the fastest among all available methods.
  - Extended the model to undirected networks, multi-source/destination networks, uncertain source/destination networks and developed new solution algorithms for these extended models.

- For network interdiction problems with nested-information
  - Proposed a game-theoretic model to address the problem where the attacker has partial information about the network state and the network may refrain from using available arcs to deceive the attacker.
  - Developed a subgame decomposition method which decomposes the multi-stage games into a sequence of one-stage subgames by exploring the nested information structure, and finds the original equilibrium strategies by solving these subgames.
  - Improved the computation efficiency exponentially for games with Markov structure. Experiments shows that our method is faster than the comparable LP approach in (von Stengel, 1996). More important, our method can handle large problems which the LP approach fails to solve.
1.4 Dissertation Overview

The rest of this dissertation is organized as follows:

Chapter 2 introduces relevant background results that provide the foundation for our results in subsequent chapters.

Chapter 3 introduces and formulates the stochastic network interdiction problem. In this chapter we develop our new solutions and provide numerical examples to illustrate the efficiency of our method, compared with previous results.

Chapter 4 extends the model in chapter 3 to undirected networks, multi-source/destination networks, uncertain source/destination networks and provides solutions to those models. In the numerical section, we implemented our method to solve interdiction problems on a real power grid, IEEE 300 Bus System, which is formulated as an undirected multi-source/destination flow network.

In chapter 5, we study the multi-stage interdiction model with information update. We solve the problem in a model-predictive approach with our extended branch and bound algorithm. In the numerical experiments, we show that our method is the fastest among all available methods. We also extend the model to undirected networks, multi-source/destination networks, uncertain source/destination networks and developed new solution algorithms for these extended models.

In chapter 6, we study dynamic interdiction games where the attacker has partial information about the network state. The problem is formulated as a zero-sum game with nested information. To find the equilibrium, we develop an algorithm, which decomposes the multi-stage games into a sequence of one-stage subgames and finds the equilibrium strategy by solving these subgames through backward and forward dynamic programming.

In chapter 7, we study games with Markov structures. By exploiting these structures, we reduce the computation of the subgame decomposition method exponen-
Initially. Numerical experiments show that our method is several orders of magnitude faster than the comparable LP approaches in (von Stengel, 1996). For larger problems where the LP approach fails to solve, our method can output high quality solutions within a short time.

In chapter 8, we summarize this dissertation and provide potential directions for future research.
Chapter 2

Background

This chapter provides background for the work in this dissertation. It includes the Sequential Approximation Algorithm for Stochastic Network Interdiction Problems in Section 2.1, the L-Shaped decomposition method in Section 2.2, the Sample Average Approximation in Section 2.3. We also introduce the extensive form games and its linear programming formulation in Section 2.4, as well as the Point-Based Value Iteration method for POMDP in Section 2.5, both of which are essential to our work on the Dynamic Games with Nested information.

2.1 Stochastic Network Interdiction

Cormican et al. (Cormican et al., 1998) studied the stochastic network interdiction problems where the attacker interdicts the network to minimize the expected max-flow after the interdiction, the outcome of which is stochastic. They developed the Sequential Approximation Algorithm to solve these problems. The model there is essentially the same as that in chapter 3, where we develop a new algorithm, 2 to 3 orders of magnitude faster than their method.

Consider a directed network $G(N, A)$ with nodes $N$ and arcs $A$, together with an identified source node $s$ and a terminal node $t$. Each arc $(i, j) \in A$ has an initial capacity $u_{ij}$. Let $(t, s)$ be an artificial return arc from $t$ to $s$ with infinite capacity and $\bar{A} := A \cup \{(t, s)\}$. Let $x_{ij}$ be the flow on arc $(i, j)$ and $x_{is}$, the flow on the artificial
Figure 2.1: The interdiction problem on a directed flow network. The numbers associated with each arc are the capacity of that arc and the probability of removing that arc if it's under attack. The defender will send flows from s to t after the attack.

Let \( x_{ts} \) be the total flow from s to t. Then the max-flow problem is

$$
\max_{x \in X} x_{ts},
$$

(2.1)

where \( x \in X \) means that \( x \geq 0 \), satisfies the conditional capacity constraints

$$
x_{ij} \leq u_{ij}, \forall (i, j) \in A
$$

(2.2)

and the flow conservation constraints

$$
\sum_{(n,i) \in A} x_{ni} - \sum_{(j, n) \in A} x_{jn} = 0, \forall n \in N.
$$

(2.3)

Notice that the maximum flow through the network can be increased by at most one unit by increasing the capacity of an arc by one unit, therefore, we have

**Lemma 2.1.1** The dual variables associated with constraints (2.2) are bounded by 1.

Denote the availabilities of arcs within the network (referred as the network state) as a \(|A|\)-dimension binary vector \( \omega \) with \( \omega_{ij} = 1 \) meaning arc \((i, j)\) is not available and
\( \omega_{ij} = 0 \) otherwise, then the max-flow problem given a network state \( \omega \) is

\[
\begin{aligned}
\left\{
\begin{array}{l}
\quad h(\omega) := \max_{x \geq 0} x_{ts} \\
\quad \text{s.t.} \quad x_{ij} \leq u_{ij}(1 - \omega_{ij}), \forall (i, j) \in A. \\
\quad \sum_{(n,i) \in \bar{A}} x_{ni} - \sum_{(j,n) \in \bar{A}} x_{jn} = 0, \forall n \in N.
\end{array}
\right.
\end{aligned}
\]

Define a penalty flow problem given a network state \( \omega \) as

\[
\begin{aligned}
\left\{
\begin{array}{l}
\quad g(\omega) := \max_{x \geq 0} x_{ts} - \sum_{(i,j) \in A} \omega_{ij} x_{ij} \\
\quad \text{s.t.} \quad x_{ij} \leq u_{ij}, \forall (i, j) \in A. \\
\quad \sum_{(n,i) \in \bar{A}} x_{ni} - \sum_{(j,n) \in \bar{A}} x_{jn} = 0, \forall n \in N.
\end{array}
\right.
\end{aligned}
\]

Theorem 2.1.2 (Cormican et al., 1998) For any \( \omega \in \{0, 1\}^{\left|A\right|}, h(\omega) = g(\omega). \)

Proof: Denote \( A^*(\omega) = \{(i, j) \in A | \omega_{ij} = 1\} \), then \( h(\omega) \) is the maximum flow problem (2.1) with the additional constraints

\[
x_{ij} \leq 0, \forall (i, j) \in A^*(\omega).
\]

(2.4)

For any \( n \in N, (i, j) \in A \), let \( \alpha_{ij}, \pi_n, \mu_{ij} \) be the dual variables corresponding to the constraints of (2.2), (2.3) and (2.4) respectively. Then the dual problem of \( h(\omega) \) is

\[
\begin{aligned}
\left\{
\begin{array}{l}
\quad hD(\omega) := \min_{\pi, \alpha, \mu} \sum_{(i,j) \in A} u_{ij} \alpha_{ij} \\
\quad \text{s.t.} \quad \pi_t - \pi_s \geq 1, \\
\quad \quad \alpha_{ij} + \pi_t - \pi_j \geq 0, \forall (i, j) \in A - A^*(\omega) \\
\quad \quad \alpha_{ij} + \pi_i - \pi_j + \mu_{ij} \geq 0, \forall (i, j) \in A^*(\omega) \\
\quad \quad \alpha_{ij}, \mu_{ij} \geq 0, \forall (i, j) \in A.
\end{array}
\right.
\end{aligned}
\]

Denote the optimal primal and dual solution to \( h(\omega) \) as \((x^*, \pi^*, \alpha^*, \mu^*)\). If we can show \((x^*, \pi^*, \alpha^*)\) is also the optimal primal and dual solution to \( g(\omega) \), then the proof
will be completed. Notice the feasible set in \( g(\omega) \) contains the feasible set in \( h(\omega) \), then \( x^* \) is feasible in \( g(\omega) \). Furthermore, by strong duality (Boyd and Vandenberghe, 2004), we have

\[
\sum_{(i,j) \in A} u_{ij}^* \alpha_{ij}^* = x_{is}^*.
\]

Therefore we just need to show the feasibility of \((\pi^*, \alpha^*)\) in the dual problem of \( g(\omega) \), which is

\[
\begin{array}{l}
gD(\omega) := \min_{\pi, \alpha} \sum_{(i,j) \in A} u_{ij} \alpha_{ij} \\
\text{s.t.} \quad \pi_t - \pi_s \geq 1, \\
\alpha_{ij} + \pi_i - \pi_j \geq 0, \forall (i,j) \in A - A^*(\omega) \\
\alpha_{ij} + \pi_i - \pi_j \geq -1, \forall (i,j) \in A^*(\omega) \\
\alpha_{ij} \geq 0, \forall (i,j) \in A.
\end{array}
\]

In fact, the dual constraints associated with \((i,j) \in A - A^*\) are identical in both \( hD(\omega) \) and \( gD(\omega) \). The remaining difference is that for \( hD(\omega) \)

\[
\pi_i^* - \pi_j^* + \alpha_{ij}^* + \mu_{ij}^* \geq 0, \forall (i,j) \in A^*(\omega); \tag{2.5}
\]

while for \( gD(\omega) \)

\[
\pi_i^* - \pi_j^* + \alpha_{ij}^* + 1 \geq 0, \forall (i,j) \in A^*(\omega) \tag{2.6}
\]

By lemma 2.1.1, we have \( \mu_{ij}^* \leq 1 \). Therefore any \((\pi^*, \alpha^*)\) satisfying (2.5) must satisfy (2.6). Therefore \((\pi^*, \alpha^*, \mu^*)\) is also the optimal primal and dual solution to \( g(\omega) \).

Denote the attacking strategy as a \(|A|\)-dimension binary vector \( \gamma \) where \( \gamma_{ij} = 1 \) means attacking arc \((i,j)\) and \( \gamma_{ij} = 0 \) otherwise. When arc \((i,j)\) is attacked, it requires resource \( c_{ij} \), and with probability \( p_{ij} \), the attack will be successful, i.e., \( \omega_{ij} = 1 \). We assume that the outcomes of attacks on different arcs will be independent, and that the interdictor has a budget of \( R \) that constraints the attack. Then the
interdiction problem is

\[ w^* := \min_{\gamma \in \Gamma(R)_b} E[h(\gamma \cdot I)], \quad (2.7) \]

where

\[ \Gamma(R)_b := \{ \gamma \mid \sum_{ij} c_{ij} \gamma_{ij} \leq R, \gamma_{ij} \in \{0, 1\}\} \]

is the set of feasible binary strategies and \( I \) is a \(|A|\)-dimension binary random vector where the probability of \( I_{ij} = 1 \) is given by \( p_{ij} \). Since \( h(\omega) \) is equal to \( g(\omega) \) for any binary \( \omega \), (2.7) is equivalent to

\[ w^{**} := \min_{\gamma \in \Gamma(R)_b} E[g(\gamma \cdot I)], \quad (2.8) \]

meaning that \( w^* = w^{**} \) and both problems have the same optimal solution.

When relaxing \( \omega \) from \( \omega \in \{0, 1\}^{|A|} \) to \( \omega \in [0, 1]^{|A|} \), we will show that \( h(\omega) \) is concave and \( g(\omega) \) is convex.

**Lemma 2.1.3** \( h(\omega) \) is concave on \( \omega \in [0, 1]^{|A|} \).

**Proof:** For any \( \omega^1, \omega^2 \in [0, 1]^{|A|} \), let \( x^s \) be an optimal solution to \( h(\omega^s), s = 1, 2 \), then we have

\[ \lambda x^1_{ij} + (1 - \lambda) x^2_{ij} = \lambda h(\omega^1) + (1 - \lambda) h(\omega^2), \quad \forall \lambda \in [0, 1] \]

Since for any \((i, j) \in A\), \( x^k_{ij} \leq u_{ij}(1 - \omega^k_{ij}), k = 1, 2 \), then

\[ \lambda x^1_{ij} + (1 - \lambda) x^2_{ij} \leq u_{ij}[1 - (\lambda \omega^1_{ij} + (1 - \lambda) \omega^2_{ij})], \]

which is the capacity constraint in \( h(\lambda \omega^1 + (1 - \lambda) \omega^2) \). Moreover, \( \lambda x^1_{ij} + (1 - \lambda) x^2_{ij} \) satisfies flow conservation since \( x^k, k = 1, 2 \) satisfy flow conservation, then \( \lambda x^1 + (1 - \lambda) x^2 \) is a feasible solution in \( h(\lambda \omega^1 + (1 - \lambda) \omega^2) \). Notice \( h(\omega) \) is maximizing over \( x \),
then

\[ h(\lambda \omega^1 + (1 - \lambda)\omega^2) \geq \lambda x_{i_1}^1 + (1 - \lambda)x_{i_2}^2 = \lambda h(\omega^1) + (1 - \lambda)h(\omega^2). \]

So \( h(\omega) \) is concave on \( \omega \in [0, 1]^{|A|} \).

\[ \text{Lemma 2.1.4} \quad g(\omega) \text{ is convex on } \omega \in [0, 1]^{|A|}. \]

\[ \text{Proof:} \quad \text{For any } \omega^1, \omega^2 \in [0, 1]^{|A|}, \lambda \in [0, 1], \text{ let } \tilde{x} \text{ be an optimal solution to } g(\lambda \omega^1 + (1 - \lambda)\omega^2), \text{ then} \]

\[ g(\lambda \omega^1 + (1 - \lambda)\omega^2) = \max_{x \in X} x_{i_1} - \sum_{i_2} (\lambda \omega_{i_1}^1 + (1 - \lambda)\omega_{i_2}^2)x_{i_2}^2 = \tilde{x}_{i_1} - \sum_{i_2} (\lambda \omega_{i_1}^1 + (1 - \lambda)\omega_{i_2}^2)\tilde{x}_{i_2}^2 \\
= \lambda (\tilde{x}_{i_1} - \sum_{i_2} \omega_{i_1}^1\tilde{x}_{i_2}^2) + (1 - \lambda)(\tilde{x}_{i_1} - \sum_{i_2} \omega_{i_2}^2\tilde{x}_{i_2}^2) \leq \lambda g(\omega^1) + (1 - \lambda)g(\omega^2). \]

The last inequality is because \( g(\omega) \) maximizes on \( \omega \) and \( \tilde{x} \) is feasible in both \( g(\omega^1) \) and \( g(\omega^2) \).

\[ \text{Theorem 2.1.5} \]

\[ \min_{\gamma \in \Gamma(R)_b} g(\gamma \cdot E[I]) \leq w^* \leq \min_{\gamma \in \Gamma(R)_b} h(\gamma \cdot E[I]). \]

\[ \text{Proof:} \quad \text{Since } h(\omega) \text{ is concave, by } \text{Jensen's Inequality (Jensen, 1906),} \]

\[ E[h(\gamma \cdot I)] \leq h(\gamma \cdot E[I]), \]

take the minimization over \( \gamma \) on both sides, we have

\[ w^* \leq \min_{\gamma \in \Gamma(R)_b} h(\gamma \cdot E[I]). \]
Similarly, one can use the convexity of \( g(\omega) \) and Jensen's Inequality to show

\[
\min_{\gamma \in \Gamma(R)_b} g(\gamma \cdot E[I]) \leq w^*.
\]

Theorem 2.1.5 states that (2.7) and (2.8) are upper and lower bounds for the original problem (2.7). Compared with problem (2.7), the size of the lower bound problem is exponentially smaller since it only considers the expected outcome. Due to strong duality, one can replace \( g(\cdot) \) with its dual in the lower bound problem \( \min_{\gamma \in \Gamma(R)_b} g(\gamma \cdot E[I]) \), which then becomes a mixed-integer linear programming (MILP) problem.

**Definition 2.1.6** A partition of \( \Omega \), denoted as \( \Phi \), is a set of subsets \( \{\phi_i\} \) of \( \Omega \), such that \( \phi_i \cap \phi_j = \emptyset \) if \( i \neq j \); and \( \cup_i \phi_i = \Omega \).

The gap between these bounds can be improved by taking finer and finer partitions on the domain of random variable \( I \), as shown next. Given \( \Phi \), define

\[
LBMIN(\Phi) := \min_{\gamma \in \Gamma(R)_b} \sum_{\phi \in \Phi} P(\phi) g(\rho(\phi) \cdot \gamma),
\]

where \( P(\phi) \) is the probability of subset \( \phi \) and

\[
\rho(\phi)_{ij} := E[I_{ij}|I \in \phi], \quad \forall (i, j) \in A
\]

Denote the approximate value of a binary strategy \( \gamma \) given \( \Phi \) as

\[
UB(\Phi, \gamma) := \sum_{\phi \in \Phi} P(\phi) h(\rho(\phi) \cdot \gamma).
\]

**Definition 2.1.7** Let \( \Phi^1, \Phi^2 \) be two partitions of \( \Omega \). \( \Phi^2 \) is said to be more refined than \( \Phi^1 \), denoted \( \Phi^1 \leq \Phi^2 \), if and only if

\[
\forall \phi_k^1 \in \Phi^1, \exists \{\phi_m^2\}_{m=1}^M \subset \Phi^2, \text{ such that } \phi_k^1 = \cup_{m=1}^M \phi_m^2
\]
Theorem 2.1.8 Let $\Phi^1, \Phi^2$ be two partitions of $\Omega$. If $\Phi^1 \leq \Phi^2$, then

$$\text{LBMIN}(\Phi^1) \leq \text{LBMIN}(\Phi^2), \quad \text{and} \quad \text{UB}(\Phi^2, \gamma) \leq \text{UB}(\Phi^1, \gamma).$$

Proof: For any $\gamma$,

$$\sum_{\phi^2 \in \Phi^2} P(\phi^2) g(\rho(\phi^2) \cdot \gamma) = \sum_{\phi^1 \in \Phi^1} P(\phi^1) \sum_{\phi^2 \subset \phi^1} \frac{P(\phi^2)}{P(\phi^1)} g(\rho(\phi^2) \cdot \gamma).$$

Because $g(\cdot)$ is convex, by Jensen’s Inequality, we have

$$\sum_{\phi^2 \in \Phi^2} P(\phi^2) g(\rho(\phi^2) \cdot \gamma) \geq \sum_{\phi^1 \in \Phi^1} P(\phi^1) g\left( \sum_{\phi^2 \subset \phi^1} \frac{P(\phi^2)}{P(\phi^1)} \rho(\phi^2) \cdot \gamma \right)$$

$$= \sum_{\phi^1 \in \Phi^1} P(\phi^1) g(\rho(\phi^1) \cdot \gamma).$$

Take the minimization over $\gamma$ on both sides, we have

$$\text{LBMIN}(\Phi^1) \leq \text{LBMIN}(\Phi^2).$$

Similarly, using the concavity of $h(\cdot)$ and Jensen's Inequality, one can show

$$\text{UB}(\Phi^2, \gamma) \leq \text{UB}(\Phi^1, \gamma).$$

To solve $\text{LBMIN}(\Phi)$, one can replace $g(\omega)$ with its dual $g_D(\omega)$ and convert $\text{LBMIN}(\Phi)$ to a MILP problem.

The idea behind the Sequential Approximation Algorithm (SAA) in (Cormican et al., 1998) is to create a sequence of finer and finer partitions until the gap between the lower and upper bounds is less than a positive tolerance $\epsilon$. SAA algorithm is summarized in Procedure (1).

There are two steps in SAA's partitioning subroutine. The first one is Selecting a cell to subdivide where the method calculates the gap for each cell of current partition
**Procedure 1 Sequential Approximation Algorithm (SAA)**

1. Let $\Phi = \{\Omega\}, U^* = \inf$ and $L^* = 0$.
2. Solve $LBMIN(\Phi)$, denote the optimal value as $L^*$ and the solution as $\gamma$.
3. if $U^* - L^* \leq \epsilon$ then
   4. Output solution $\gamma^*$ and terminate.
5. end if
6. Evaluate $U' := UB(\Phi, \gamma)$.
7. if $U' < U^*$ then
   8. Update $\gamma^* \leftarrow \hat{\gamma}$ and $U^* \leftarrow U'$.
9. end if
10. if $U^* - L^* \leq \epsilon$ then
11. Output solution $\gamma^*$ and terminate.
12. end if
13. Refine partition $\Phi$ according to partition procedure described below and go to Step 2.

with current binary solution $\hat{\gamma}$

$$D(\phi, \hat{\gamma}) := p(\phi)[h(p(\phi) \cdot \hat{\gamma}) - g(p(\phi) \cdot \hat{\gamma})], \forall \phi \in \Phi,$$

and selects the cell with the largest $D(\phi, \hat{\gamma})$, denoted as $\hat{\phi}$. The second step is Subdividing on arcs within the cell. Denote the set of arcs that do not have fixed outcomes within $\hat{\phi}$ as

$$A(\hat{\phi}) := A - \{(i, j) | I_{ij} \equiv 0, \text{ or } I_{ij} \equiv 1, \forall I_{ij} \in \hat{\phi}\}.$$

Initially $A(\hat{\phi}) = A$ since $\hat{\phi} = \Omega$ is the unique cell in the initial partition $\{\Omega\}$ and $\Omega$ has no fixed outcome on any arc. The partition on $\hat{\phi}$ is restricted in $A(\hat{\phi})$. The method estimates the gaps as if splitting $\hat{\phi}$ on any arc $(i, j)$ in $A(\hat{\phi})$. Let $\phi_{ij}^0(\hat{\phi}), \phi_{ij}^1(\hat{\phi})$ be the two new cells resulting from splitting $\hat{\phi}$ on arc $(i, j)$, i.e.,

$$\phi_{ij}^0(\hat{\phi}) := \phi \bigcap \{I_{ij} = 0\}, \quad \phi_{ij}^1(\hat{\phi}) := \phi \bigcap \{I_{ij} = 1\}.$$

then the new gap between bounds in $\hat{\phi}$ is

$$D_{ij} := D(\phi_{ij}^0(\hat{\phi}), \hat{\gamma}) + D(\phi_{ij}^1(\hat{\phi}), \hat{\gamma}).$$
Let \((i^*, j^*)\) be the arc that has the minimum \(D_{ij}\). Then SAA splits \(\hat{\phi}\) on arc \((i^*, j^*)\), which yields the new partition for the next iteration. Notice in each iteration, SAA only splits one old cell and keep others the same. Therefore the new partition has one more cell than the one in the previous iteration.

Our algorithm developed in chapter 3 improves SAA in two aspects: (1) solving the approximation problem and (2) refining partitions, which will be discussed in details in section 3.2.

### 2.2 L-Shaped Decomposition

Problem (2.7) has the structure of a two-stage decision problem, where the first stage is the binary strategy \(\gamma\), which is made before the observation of the random outcome and the second one is the network flows after the outcome. Bender’s Decomposition (Benders, 1962), or the L-Shaped method (Slyke and R.J.Wets, 1969) when it applies to stochastic linear programming problems, is suitable in solving linear problems with such stochastic structure. This method is a basic subroutine in our algorithms developed for both the one-stage problem and the two-stage problem in chapter 3 to chapter 5.

Consider the stochastic network interdiction problem of (2.8)

\[
\min_{\gamma \in \Gamma(R)} E[g(\gamma \cdot I)] = \sum_I P(I)g(\gamma \cdot I),
\]

where \(P(I)\) is the probability of scenario \(I\). The method decomposes (2.11) into a master problem (2.12) and a set of subproblems (2.13)

\[
\min_{(\gamma, z) \in F} z,
\]
where $F := \Gamma(R)_b \otimes R$ with $\otimes$ as Cartesian product;

$$g(\gamma \cdot I) := \max_x x_{ts} - \sum_{(i,j) \in A} \gamma_{ij} I_{ij} x_{ij}$$

(2.13)

where $x$ satisfies the flow conservation constraints (2.3) and the capacity constraints (2.2). In the master problem (2.11), the initial feasible set is $F = \Gamma(R)_b \otimes R$. The L-Shaped method solves the master problem and checks its solution with subproblems (2.13). Based on the optimal solutions of the subproblems, it constructs an additional effective constraint for the master problem. By repeating above procedure iteratively, the L-Shaped method finds an optimal solution which makes all subproblems feasible and bounded. The details are shown in Procedure 2 where $\epsilon$ is a positive tolerance.

**Procedure 2 L-Shaped Decomposition**

1: Solve (2.12), denote the optimal solution as $(\gamma^*, z^*)$.
2: For any $I$, solve (2.13), let $x^I$ be the optimal solution.
3: Check the inequality $\sum_I P(I) x^I_{ts} - \sum_{(i,j) \in A} I_{ij} \gamma^*_i x^I_{ij} \leq z^* + \epsilon$.
4: if the inequality is false then
5: Update the feasible set in the master problem with the additional constraint
6: Go to Step 1.
7: end if
8: Output $\gamma^*$ as the optimal solution, $z^*$ as the optimal value of (2.11).

By replacing $\sum_I P(I) g(\gamma \cdot I)$ with $z$ in the objective function and adding a constraint $z \geq \sum_I P(I) g(\gamma \cdot I)$, problem (2.11) can be transformed to an equivalent problem. One can find that any solution in the feasible set of the transformed problem must be an solution in $F$ and satisfies the additional constraints in iterations. Since both minimization problems have the same objective function $z$, the one with larger feasible set is the lower bound of the other problem. Therefore, we have

**Proposition 2.2.1** In each iteration, $z^*$ is the lower bound of (2.11).
Denote $U$ as $U := \sum_I P(I) [x^I_{ts} - \sum_{(i,j) \in A} I_{ij} \gamma^*_{ij} x^I_{ij}]]$, because $\gamma^*_{ij}$ is a feasible solution to a minimization problem (2.11) and $U$ is its objective value, then

**Proposition 2.2.2** In each iteration, $U$ is the upper bound of (2.11).

Since $z^*$ is the lower bound and $U$ is the upper of the optimal value of (2.11), then

**Proposition 2.2.3** If $U < z^* + \epsilon$, then $z^*$ is the optimal value of (2.11) with tolerance $\epsilon$ and $\gamma^*$ is the corresponding solution.

An alternative decomposition is to decompose the original problem into a set of subproblems of (2.13) and following master problem

\[
\min_{(\gamma, \{z^I\}) \in \tilde{P}} \sum_I P(I) z^I, \tag{2.14}
\]

where $\tilde{P} := \Gamma(R)_b \otimes R^n$ with $n$ is the total number of subproblems. In each iteration, let the optimal solution of the master problem be $(\gamma^*, \{z^*I\})$. For each subproblem, check whether

\[
x^I_{ts} - \sum_{(i,j) \in A} I_{ij} \gamma^*_{ij} x^I_{ij} \leq z^*I + \frac{\epsilon}{n}
\]

is valid. Once an invalid inequality is found, add the following constraint to the master problem and start the next iteration

\[
x^I_{ts} \leq \sum_{(i,j) \in A} (I_{ij} x^I_{ij}) \gamma_{ij} + z^I.
\]

With similar deduction as for the original L-Shaped decomposition, one can show that the alternative decomposition also yields the optimal solution to Problem (2.11).

The inequality that we check within each iteration is called the **Optimality Inequality**, which checks whether the current solution of the master problem is (approximately) the optimal solution of the original problem. When implemented for general stochastic programming problems, there is an additional inequality for each subproblem, called the **Feasibility Inequality**, which checks whether the subproblem
is feasible based on the current solution of the master problem. However, in our case, (2.13) is always feasible and bounded for any values of $\gamma^* \in \Gamma(R)$. Therefore we can omit the feasibility check in our implementation.

Note that in the decomposition method, $\gamma$ and $I$ are not required to be binary. In fact, in some implementations in the rest of this dissertation, $\gamma$ and $I$ are fractional variables.

2.3 Sample Average Approximation

The stochastic interdiction problem formulated before is essentially a two-stage stochastic mixed-integer program (SMIP) having discrete first-stage variables (the binary interdiction strategy), and continuous second-stage variables (the flows on arcs). Sanchez and Wood (Sanchez and Wood, 2006) proposed a “BEST” algorithm for general SMIP problems which uses Monte Carlo techniques to deal with the difficulty arising from the problem size, due to the exponential number of outcomes in stochastic problems. Janjarassuk and Jeff (Janjarassuk and Linderoth, 2008) tailored this method to stochastic network interdiction problems.

Consider a stochastic network interdiction problem

$$\min_{\gamma \in \Gamma(R)} E[h(\gamma \cdot I)] = \sum_I P(I)g(\gamma \cdot I). \quad (2.15)$$

Since the number of outcomes of $I$ grows exponentially with the number of arcs, the size of the problem is exponentially large. The Sample Average Approximation Approach (SAM) first solves $M$ approximation problems. Each of them considers $N$ independently sampled scenarios. For problem $m, m = 1, \ldots, M$, denote the sampled scenarios as $I^m_n, n = 1, \ldots, N$, then the approximation problem is

$$\min_{\gamma \in \Gamma(R)} \frac{1}{N} \sum_{n=1}^N g(\gamma \cdot I^m_n). \quad (2.16)$$
With the decomposition techniques introduced before, SAM solves this problem as a mixed-integer programming (MIP) problem. Denote its solution and optimal value as $\gamma^m, L^m$ respectively. Let $\gamma^*, w^*$ be the optimal solution and the optimal value of (2.15).

**Proposition 2.3.1** Let $\bar{L} := \frac{1}{M} \sum_{m=1}^{M} L^m$, which is a random value due to the random samples of $I_n^m$, then

$$E[\bar{L}] \leq w^*.$$  

**Proof:** By definition, we have

$$E[\bar{L}] = \frac{1}{M} \sum_{m=1}^{M} E[\frac{1}{N} \sum_{n=1}^{N} g(\gamma^m \cdot I_n^m)].$$

Notice that $\gamma^m$ is the optimal solution to (2.16), then

$$E[\frac{1}{N} \sum_{n=1}^{N} g(\gamma^m \cdot I_n^m)] \leq E[\frac{1}{N} \sum_{n=1}^{N} g(\gamma^* \cdot I_n^m)] = E[g(\gamma^* \cdot I)],$$

where the last equation is because $I_n^m$ are i.i.d. Therefore $E[\bar{L}] \leq E[g(\gamma^* \cdot I)] = w^*$.

After solving these approximation problems, SAM has $M$ solutions, denoted as $\{\gamma^m\}_1^M$. Then it evaluates the performances of these solutions by a set of random sampled scenarios $\{I_e\}_{e=1}^E$. For any $\gamma^m, m = 1, \ldots, M$, define its approximation objective value based on $\{I_e\}_{e=1}^E$ as

$$U^m := \frac{1}{E} \sum_{e=1}^{E} g(\gamma^m \cdot I_e).$$

**Proposition 2.3.2** Let $\bar{U} := \frac{1}{M} \sum_{m=1}^{M} U^m$, which is a random value due to the random samples of $I_e$, then

$$E[\bar{U}] \geq w^*.$$
Proof: By definition, 

\[ E[\bar{U}] = \frac{1}{ME} \sum_{m=1}^{M} \sum_{e=1}^{E} E[g(\gamma^m \cdot I_e)]. \]

Notice that \( I_n^m \) are i.i.d, and \( \gamma^* \) is the optimal solution to (2.15), then

\[ E[g(\gamma^m \cdot I_e)] = E[g(\gamma^m \cdot I)] \geq E[g(\gamma^* \cdot I)]. \]

therefore \( E[\bar{U}] \geq E[g(\gamma^* \cdot I)] = w^* \).

SAM selects the solution with the minimum \( U^m \) as the output. In SAM, the samples in \( \{I_n^m\}_{n,m=1}^{N,M} \) can be reused in \( \{I_e\}_{e=1}^{E} \) and \( E \) is usually much larger than \( N \).

SAM provides a lower bound \( \bar{L} \) and the upper bounds \( \bar{U} \) in statistical sense. For either bound, it can also estimate the range of the bound with a given confidence level according to the Student’s t-test. However, due to its random nature, it’s possible that \( \bar{L} > \bar{U} \). Therefore we would use \( \bar{L}, \bar{U} \) as measures of the estimation quality rather than a stopping criteria.

We will show with numerical results that our algorithm not only guarantees optimality but is also faster than the Sample Average Approximation.

2.4 Games in Extensive Form

From a game-theoretic approach, the network interdiction problem can be formulated as a zero-sum game between the attacker and the network. To better represent the dynamics in the multi-interaction problems, we use extensive form games.

Generally speaking, a game consists of three elements: players, actions and payoffs. Payoffs are functions of the game’s outcome (or ending), which is a combination of players’ actions that leads to the outcome. Each player tries to maximize its payoff by choosing its actions (called strategy). A game can be single-stage where all players act once at the same time, or multi-stage where players act at different times.
An extensive form is a directed tree where a node represents a state of the game (denoted by a history of players' past actions) while an edge represents a player's action on that state (the starting node) which leads the game develop into the next state. The leaf nodes are the outcomes of the game, which are associated with players' payoffs. Each non-leaf node (also called a decision node) belongs to exactly one player, who will act on that node.

**Definition 2.4.1** In an extensive form game, a sequence, defined by a node of the game tree, is a string of actions from the root to that node. A subsequence of player $X$, is the subsequence (of that node) that contains all the actions belonging to player $X$.

**Definition 2.4.2** A play is a sequence corresponding to a leaf node in the game tree, i.e., a play represents a path of the game from start to end.

If the game has $n$ players, then each play is associated with an $n$-dimension tuple, representing the payoffs to these players. If the randomness (if any) within a game is considered as an action of player Nature, then Nature has no payoff.

**Definition 2.4.3** An information set is a set of non-leaf nodes in the extensive form, such that

- All nodes within the set belong to the same player;
- That player can NOT tell the difference between nodes within that set;
- That player can distinguish nodes within that set from its other nodes outside that set.

For the class of games considered in this thesis, information sets will arise because of partial knowledge of players' past actions. Because a player can not tell the difference of nodes within an information set, its actions on these nodes must be the same. Therefore, we say a player acts on an information set instead of an individual node.

Fig. 2.4 is the extensive form of a simple stochastic network interdiction game, where the underlying network has two arcs that are uncertain and interdictable. First,
Nature (the top square) selects the network states (4 solid nodes in the second line). The availabilities of arcs are represented by two binary digits with 1 being available and 0 otherwise. For example, 01 is the network state where only the second arc is available. Nature's action is denoted as $s_{ij}^0$, $ij = 00, 01, 10, 11$ where $s_{ij}^0$ means Nature selects network state $ij$. Then the attacker acts, represented by solid arrows starting from the nodes in the second line. Since the attacker does not know the exact network state, its information set (the ellipse in the second line) contains all $s_{ij}^0$. The attacker can detect either the first arc (left arrow, denoted as $a_1^i$) or the second arc (right arrow, denoted as $a_2^i$). Then the network acts, represented by dashed arrows. Notice the network is assumed to know the network state, therefore its information sets all contain one single node. It can either send a flow on the detected arc if it’s available (left arrow, denoted as $d_1^1$) or no flow (right arrow, denoted as $d_2^0$). For some network states when the detected arc is available, no flow means the network refrains from using that arc. At last the attacker acts again. Its information sets are the ellipses in the fourth line. Since it does not know the network state, it can not tell the difference between nodes within each ellipse. The corresponding substrings for these information sets (from left to right) are $[a_1^1, d_6^0], [a_1^2, d_7^0], [a_2^1, d_8^0], [a_2^2, d_9^0]$ respectively. The attack is represented by solid arrows starting from the fourth line. The attacker can attack either the first arc (left arrow, denoted as $a_1^3$) or the second arc (right arrow, denoted as $a_2^3$). Then the game ends. A play of the game is a leaf node in the tree, represented by a sequence of actions. For example, all nodes in the fifth line are plays, where the second one can be represented by $[s_{01}^0, a_1^1, d_6^0, a_2^3]$.

Let $I^{X,s}$ denote the information sets of player $X$, where $s$ is an index of the information sets.

**Definition 2.4.4** A strategy of a player $X$, denoted as $S^X$, is a function of player $X$'s information sets such that it maps each $I^{X,s}$ into an admissible action for player $X$ at any of the nodes in $I^{X,s}$. 
Figure 2.2: The extensive form of a network interdiction game.

Strategies can be in one of three forms: pure strategies, behavior strategies and mixed strategies. In pure strategies, players act deterministically at all information sets.

**Definition 2.4.5** A *mixed strategy* is a strategy that the player selects a pure strategy randomly according to a probability distribution over the set of pure strategies.

In behavior strategies, players act randomly with a given distribution on possible actions at each information set. Let \( A(I_{X,s}) \) be the set of possible actions on the information set \( I_{X,s} \) and \( D(I_{X,s}) \) be the set of probability distributions on \( A(I_{X,s}) \).

**Definition 2.4.6** A *behavior strategy* of player \( X \), denoted as \( B_X \), is a strategy such that for any \( I_{X,s} \), \( B_X(I_{X,s}) \) is an element in \( D(I_{X,s}) \) with \( B(a)(I_{X,s}) \) for any \( a \in A(I_{X,s}) \) being the probability of taking action \( a \) at \( I_{X,s} \).

The difference between mixed strategies and behavior strategies is that in the former, players' actions are correlated between information sets while in the latter they are independent.

Players' strategies affect the development of a game, including its final endings and the associated payoffs to these players.

**Definition 2.4.7** The total (expected if any randomness is involved) payoff of a player \( X \) is a function of all players' strategies, which can be denoted as \( p_X(S^1, S^2, \ldots, S^n) \), where \( S^n \) is player \( n \)'s strategy.

Given the strategies of all players, one can calculate the distribution of the plays in a game. Then the total payoff to a player is the expected payoff based on this distribution and the player's payoffs in all plays.
Definition 2.4.8 A Nash equilibrium is a set of strategies $S^1, S^2, \ldots, S^n$, such that no player gets a better payoff by unilaterally changing its strategy. That is, for any player $X = 1, \ldots, n$,

$$p^X(S^1, \ldots, S^X*, \ldots, S^n*) \geq p^X(S^1, \ldots, S^X, \ldots, S^n*), \forall S^X.$$

There is much work in the game theory literature on finding Nash equilibria. von Neumann and Morgenstern (von Neumann and Morgenstern, 1944) showed that

Theorem 2.4.9 For any zero-sum game with a finite set of actions, there exists a Nash equilibrium in terms of mixed-strategies.

Definition 2.4.10 Two strategies of a player are called realization equivalent if and only if for any fixed strategies of other players, both strategies define the same probabilities for reaching nodes in the extensive form tree.

If two strategies are realization equivalent, for any strategies of other players, they yield the same distribution of the outcomes of the game, thus the same payoff values to players. Obviously any pure strategy is realization equivalent to a special behavior strategy where at each state the player selects one action with 100% probability. Given a combination of all players' behavior strategies, one can calculate the distribution of outcomes and find a convex combination of all player's pure strategies that yields the same distribution. In this sense, any behavior strategy is realization equivalent to a mixed strategy. The inverse is not necessarily true for general games.

Definition 2.4.11 A player is said to have Perfect Recall if and only if it knows what it knew (this feature is called perfect memory) and what it did in previous stages.

The feature of Perfect recall contains two meanings (Okada, 1987): each player knows what it knew and what it did in previous stages. Most games fall into the category where all players have perfect recall, including the problems that we’re studying. For games with such features, we have following result
Theorem 2.4.12 (Kuhn, 1950) For a game where all players have perfect recall, any mixed strategy is realization equivalent to a behavior strategy.

By theorem 2.4.12, one can find an equilibrium in terms of behavior strategies rather than mixed strategies.

Consider a $N$ player game, given players' behavior strategies $B^n, n = 1, \ldots, N$, for any sequence $s$, denote the sequence of player $n$ as $s_n$ where $s_n \subset s$, then the probability of $s$ is

$$\prod_{n=1}^{N} \prod_{a \in s_n} B^n_a(I^n_r),$$

where $I^n_r$ is the information set where player $n$ takes action $a \in s_n \subset s$, thus it's on the path of sequence $s$.

Let $P$ be the set of plays in the game, and $c^n(p)$ for any $p \in P$ be player $n$'s payoff when the game ends at $p$, then player $n$'s objective is to maximize

$$\sum_{p \in P} \prod_{k=1}^{N} \prod_{a \in s_k} B^k_a(I^k_r)c^k(p)$$ (2.17)

over its behavior strategy $B^n$.

Notice (6.3) is a complex function of $B^n$. von Stengel (von Stengel, 1996) developed the sequence form formulation for games with perfect recall, where finding the equilibrium of a two-person zero-sum game is equivalent to solving a linear program (LP) problem with the size proportional to the size of the extensive form tree. We will show this result in the rest of this section.

Definition 2.4.13 A realization plan of player $n$, denoted as $r^n$, is a mapping from the set of its sequences $S^n$ to $\mathbb{R}^+$, such that

$$r^n(s_0^n) = 1; \sum_{a: s_n = [s'_n, a]} r^n(s_n) = r^n(s'_n), \forall s'_n \in S^n$$ (2.18)

where $s_0^n$ is player $n$'s first information set(s) and $s_n = [s'_n, a]$ means the sequence $s_n$ is extended by sequence $s'_n$ with $a$. 
Given player $n$'s behavior strategy $B^n$, one can calculate $r^n(s)$ as,

$$r^n(s) := \prod_{a \in \mathcal{A}} B^n_a(I^n_a), \forall s \in S^n. \quad (2.19)$$

For games with perfect recall, within each information set, there is a common subsequence corresponding to that player's previous actions reaching to (all nodes of) that set. Denote $I^n_{a,s_n}$ as an information set with such subsequence $s_n$, then by (2.19),

$$\sum_{a \in A(I^n_{a,s_n})} r^n([s_n, a]) = \sum_{a' \in A(I^n_{a,s_n})} B^n_a(I^n_{a,s_n}) \prod_{a \in S_n} B^n_a(I^n_{a'}) = r^n(s_n).$$

Since the above equation is true for any $I^n_{a,s_n}$, then $r^n(s)$ calculated by (2.19) is a realization plan. That is, in a perfect recall game, given a player's behavior strategy, one can calculate the corresponding realization plan by (2.19). The reverse is not true for general games, because an information set in a non-perfect-recall game may not have the common subsequence representing the player's previous actions to that set. However, for games with perfect recall, we have

**Theorem 2.4.14** (von Stengel, 1996) For games with perfect recall, any realization plan arises from a suitable behavior strategy.

**Proof:** For any realization plan $r^n$ satisfying (2.18), we can define $B^n$ recursively such that for any $s'_n \in S^n$

$$\begin{align*}
\text{if } r^n(s'_n) > 0, & \quad B^n_a(I^n_{a,s'_n}) := \frac{r^n(s_n)}{r^n(s'_n)}, \forall a \in A(I^n_{a,s'_n}); \\
\text{otherwise } & \quad B^n_a(I^n_{a,s'_n}) \geq 0 \text{ with } \sum_{a \in A(I^n_{a,s'_n})} B^n_a(I^n_{a,s'_n}) = 1.
\end{align*}$$

By the above formula and (2.18), for any $s' \in S^n$, we have

$$\sum_{a \in A(I^n_{a,s'})} B^n_a(I^n_{a,s'}) = \sum_{a \in A(I^n_{a,s'})} \frac{r^n(s_n)}{r^n(s'_n)} = 1.$$
Notice $B_n^*(I^{n,s'}) \geq 0$, therefore $B_n^*(I^{n,s'})$ is a distribution on $I^{n,s'}$. Since this is true for all $I^{n,s'} \in I^n$, then $B_n^*$ is a behavior strategy.

By theorem 2.4.14, finding the equilibrium behavior strategies is equivalent to finding the corresponding realization plans since with the realization plans, one can construct the corresponding behavior strategies via the equations within theorem 2.4.14.

In a two-person zero-sum game, denote the players as $X,Y$ and $c$ as the payoff to player $X$. Let $x,y$ be player $X$'s realization plan and player $Y$’s realization plan respectively, with their probability conservation constraints denoted as $E_x = e$ and $F_y = f$, then player $X$ is to

$$\max_{x \in X} \min_{y \in Y} \sum_{p \in P} x(p)y(p)c(p), \quad (2.20)$$

and player $Y$ is to

$$\min_{y \in Y} \max_{x \in X} \sum_{p \in P} x(p)y(p)c(p), \quad (2.21)$$

where $X := \{x|x \geq 0, E_x = e\}, Y := \{y|y \geq 0, F_y = f\}$ and $P$ is the set of the game’s play. Notice the objective function is a bilinear function on $x, y$, we can denote it as $x'C_y$ with $C$ derived from $c(p)$ for all $p \in P$. By replacing the inner optimization problems with their duals in (2.20) and (2.21) respectively, (2.20) and (2.21) can be converted to LP problems, which have the same optimal value by strong duality. Then we have

**Theorem 2.4.15** (von Stengel, 1996) The equilibria of a zero-sum game in extensive form with perfect recall are the optimal primal and dual solutions of a linear program whose size, in sparse representation, is linear in the size of the game tree.
2.5 Point-Based Value Iteration

Point-Based Value Iteration (PBVI) is an approximation method for Partially Observable Markov Decision Process (POMDP), which is a generalized Markov Decision Processes with partially observed states of the system. A POMDP consists the following elements

- $S$: the (finite) set of states,
- $A$: the set of discrete actions,
- $O$: the set of observations providing incomplete/noisy information,
- $\mathcal{B}^0$: the distribution on $S$. For any $s \in S$, $\mathcal{B}^0(s)$ is the probability of initial state $s$.
- $T$: the conditional probability of $s^{t+1}$ given $s^t$ and $a^t$.
  \[ T(s, a, s') := \text{Prob}(s^{t+1} = s'|a^t = a, s^t = s). \]
- $\Omega$: the conditional probability of observing $o^t$ given $s^t$ and $a^t$.
  \[ \Omega(o, s, a) := \text{Prob}(o^t = o|a^t = a, s^t = s). \]
- $R$: the reward given state $s$ and action $a$, $R(s, a)$.
- $\gamma$: the discount factor of the reward.

With a finite number of stages $T$, a POMDP can be taken as an extensive form game between a decision maker and Nature, who controls the randomness in that process. The payoff to the decision maker is the (discounted if any) sum of periodic rewards $R(s, a)$ and the decision maker tries to maximize this payoff. A sequence in a POMDP is a string of states $s^t$, actions $a^t$ and observations $o^t$ before that node. Since the decision maker can not see the state directly, its information set before taking action $a^t$ is

\[ I^t := [a^1, o^1, \ldots, a^{t-1}, o^{t-1}] \].
A strategy in a POMDP is a mapping from these information sets $I^t$ into an admissible action at these sets. Since the number of the information sets grows exponentially with the number of stages, the number of possible mappings is very large, which is called the curse of history. To handle this difficulty, it has been shown that there is a sufficient statistic, called the belief state, defined as the distribution over $S$ conditioned on previous observations and actions

$$b^t := Pr(s^t|b^0, a^0, o^0, \ldots, o^{t-1}, a^{t-1}, o^t),$$

which can be updated to incorporate the latest action $a$ and observation $o$

$$b'(s') := \eta \Omega(o, s', a) \sum_{s \in S} T(s, a, s') b^{-1}(s)$$

where $\eta$ is a normalizing factor that makes $b'(s')$ as a probability distribution. Since belief states are sufficient statistics, the strategy of a POMDP at time $t$ can be restricted to be a function of the current belief state. Notice the dimension of the belief state is constant over time, the strategy of a POMDP becomes trackable. Given the strategy, one can derive the realization probabilities of all plays. Then the total payoff to the decision maker is the expected payoffs of all plays according to that distribution. Let $V^t(b)$ be the expected payoff function given the current belief is $b$ and the sequential strategies are all optimized, then this value function can be defined recursively by

$$V^t(b) := \max_{a^t} \sum_{s \in S} R(s, a)b(s) + \gamma T(b, a, b')V^{t+1}(b).$$

Denote the optimal solution of the above problem as $a^t$, then the optimal strategy maps the belief state $b$ into action $a^t$ at time $t$.

**Theorem 2.5.1** (Smallwood and Sondik, 1973) For POMDPs with discrete-time and finite actions, observations and horizon, the optimal payoff function is a piecewise-
linear, convex function on the belief of the current state.

Given a belief state $b$ at current stage $t$ and the set of support vectors $Q^{t+1}$ of the next stage value function $V^{t+1} \cdot$, then the value function of current stage is

$$V^t(b) = \max_{a \in A} \left[ \sum_{s \in S} R(s, a) b(s) + \gamma \sum_{o \in O} \max_{q \in Q^{t+1}} \sum_{s' \in S} T(s, a, s') \Omega(o, s', a) q(s') b(s) \right].$$

(2.22)

Sondik et al. (Smallwood and Sondik, 1973) proposed a single-pass value-iteration method, which decomposes the multiple stages into a set of single-stage decision problems and then implements backward induction to find the set of support vectors stage by stage. This method was further improved by numerous researches in (Monahan, 1982), (Kaelbling et al., 1998), (Cheng, 1988). To improve these algorithms' efficiency, Pineau et al. (Pineau et al., 2003) argued that the inefficiency is due to the curse of history, i.e. the number of histories grow exponentially with the number of stages. They proposed a Point-Based Value Iteration solution which approximately estimates the payoff function by selecting a small set of representative belief points and then tracking the value and its derivative for those points only. Procedure 3 summarizes the PBVI algorithm. In this method, Pineau et al. (Pineau et al., 2003) provided an error bound for the approximation solution, which depends on the gap between the maximum and minimum rewards, and the maximum distance to the nearest neighbor in the belief space.

In chapter 6, we will extend this algorithm for the solution of zero-sum games with nested imperfect information.
Procedure 3 PBVI to evaluate $Q^t$ based on $Q^{t+1}$

Select a finite set of beliefs $B := \{b_i\}_{i=0}^n$.

Initiate $Q^t \leftarrow \emptyset$.

for each $b_i$ do

Solve (2.22), denote the optimal action as $a_i$, and the support vector in $Q^{t+1}$ corresponding to each $o$ as $q_{i,o}^{t+1}$.

Calculate the support vector $q_i$ for $b_i$ by

$$q_i(s) := R(s, a_i) + \gamma \sum_{o \in O} \sum_{s' \in S} T(s, a_i, s') \Omega(o, s', a_i) q_{i,o}^{t+1}(s'), \forall s \in S.$$ 

Add $q_i$ to $Q$.

end for

Output $Q$. 

Chapter 3

Branch and Bound Algorithms for Stochastic Network Interdiction Problems

We consider a stochastic network interdiction problem where an attacker chooses a set of arcs to attack with limited resources, to minimize the maximum flow that the network will be able to support after the attack. The outcome of the attack is random and binary, i.e., each arc being attacked is either removed or remains in the network with full capacity according to a given probability. The sequence of actions are shown in Fig. 3.

We assume that the outcomes are independent across arcs. This problem is essentially the same as that in (Cormican et al., 1998). Borrowing the idea of sequential approximation bounds in their paper, we develop a new solution method that integrates the branch and bound techniques with increasingly accurate lower bounds. Numerical experiments show that our method is significantly faster than previous approaches, including the Sequence Approximation Algorithm proposed in (Cormican et al., 1998).

![Diagram](image)

**Figure 3.1:** The sequence of actions of the one-stage interdiction problem.
3.1 Problem Formulation

Consider a directed network $G(N, A)$ with nodes $N$ and arcs $A$, together with an identified source $s$ and a destination $t$. Each arc $(i, j) \in A$ has an initial capacity $u_{ij}$. Let $(t, s)$ be an artificial return arc from $t$ to $s$ with infinite capacity and $\bar{A} := A \cup \{(t, s)\}$. Denote a network state $\omega$ to be a binary vector on the number of arcs, where $\omega_{ij} = 1$ indicates arc $(i, j)$ is broken and cannot be used. Let $x_{ij}$ be the flow on arc $(i, j)$ and $x_{ts}$, the flow on the artificial arc $(t, s)$, can be taken as the total flow from $s$ to $t$. Then the max-flow problem for condition $\omega$ is

$$h(\omega) := \max_{x \in X(\omega)} x_{ts}$$

(3.1)

where $x \in X(\omega)$ means that $x \geq 0$, satisfies the \textit{conditional capacity constraint}

$$x_{ij} \leq u_{ij}(1 - \omega_{ij}), \forall (i, j) \in A$$

(3.2)

and the \textit{flow conservation constraints}

$$\sum_{(n,i) \in \bar{A}} x_{ni} - \sum_{(j,n) \in \bar{A}} x_{jn} = 0, \forall n \in N.$$  

(3.3)

The optimal network action based on observed state $\omega$ will be to send flow along the solution of (3.1).

Let $\gamma$ be an interdiction strategy, which is a $|A|$-dimension binary vector with $\gamma_{ij} = 1$ means attacking arc $(i, j)$ and $\gamma_{ij} = 0$ otherwise. When arc $(i, j)$ is attacked, it requires resources $C_{ij}$, and with probability $p_{ij}$, the attack will be successful, i.e. $\omega_{ij} = 1$. We assume that the outcomes of attacks on different arcs will be independent, and that the interdictor has a budget of $R$ that constraints the attack. Denote $\Gamma(R)_b$ as the set of feasible binary strategies satisfying the budget constraint, and $\Gamma(R)$ as
its convex hull, i.e.,

$$\Gamma(R) := \{ \gamma | \sum_{ij} c_{ij} \gamma_{ij} \leq R, 0 \leq \gamma_{ij} \leq 1 \}$$

Construct an auxiliary random binary vector $I$ corresponding to the arc survival outcomes under the assumption that every arc in $A$ is attacked, so $I_{ij} = 1$ means the interdicted arc would be broken if attacked and $I_{ij} = 0$ otherwise. Let $\Omega := \{0, 1\}^{|A|}$ be the space of $I$ and $P(I)$ the probability of $I$, which is

$$P(I) = \prod_{(i,j) \in A} (1 - p_{ij})^{1-I_{ij}} p_{ij}^{I_{ij}}, \forall I \in \Omega.$$ 

The availability of arc $(i, j)$ after interdiction is given by $I_{ij} \gamma_{ij}$. Then the optimization problem for the attacker is

$$J := \min_{\gamma \in \Gamma(R)} \sum_{I \in \Omega} P(I) h(I \cdot \gamma), \quad (3.4)$$

where $I \cdot \gamma$ is the element-wise product of $I$ and $\gamma$.

There are several difficulties in solving Problem (3.4). First, the problem is combinatorial in size, where the number of possible discrete attacks grows exponentially with the number of arcs. Second, even if one relaxed the binary constraint, i.e., replacing $\Gamma(R)_b$ with $\Gamma(R)$, the resulting objective is the sum of concave functions, so minimization is again combinatorially complex. Finally, the evaluation of any strategy requires a summation over an exponential number of outcomes, which would require solving an exponential number of max-flow subproblems $h(I \cdot \gamma)$. We will develop a solution approach that addresses these issues in the next section.
3.2 Modified Branch and Bound Method

Our method solves Problem (3.4) via a branch and bound approach with increasingly accurate approximations. The basis for this approach is an alternative representation of the max-flow problem, developed in (Cormican et al., 1998).

For any $\omega \in [0,1]^{|A|}$, define the penalty problem

$$g(\omega) := \max_{x \in X} x_{ts} - \sum_{(i,j) \in A} \omega_{ij} x_{ij},$$

where $x \in X$ means $x \geq 0$, satisfies the flow conservation constraints (3.3) and the capacity constraints (unrelated to $\omega$)

$$x_{ij} \leq u_{ij}, \forall (i,j) \in A.$$  (3.6)

Compared to $h(\omega)$, $g(\omega)$ moves $\omega$ from the capacity constraints to the objective function as a penalty. Therefore the feasible set $X(\omega)$ in $h(\omega)$ is a subset of the feasible set $X$ in $g(\omega)$, which is independent from $\omega$. Next we will explore the properties of $h(\omega), g(\omega)$ as well as the relationship between them, which form the foundation of the approximation in our method.

Theorem 3.2.1 is a restatement of Theorem 2.1.2, which is proved by duality in (Cormican et al., 1998). Here we provide a direct proof, which is more intuitive and simpler.

**Theorem 3.2.1** $h(\omega) = g(\omega)$, for any $\omega \in \{0,1\}^{|A|}$.

**Proof:** For any $x \in X(\omega)$, for any $(i,j)$ it's either $\omega_{ij} = 0$ or $x_{ij} = 0$ due to capacity constraint, then we have $x_{ts} - \sum_{(i,j) \in A} \omega_{ij} x_{ij} = x_{ts}$. Therefore when $x \in X(\omega)$, problems $h(\omega)$ and $g(\omega)$ have the same objective value. Notice $X(\omega) \subset X$, now we just need to show that $g(\omega)$ always has an optimal solution in $X(\omega)$. 
Let $x^*$ be an optimal solution to $g(\omega)$, such that $x^* \notin X(\omega)$, i.e.

$$A(x^*) := \{(i, j) | (i, j) \in A, \omega_{ij} = 1 \text{ and } x^*_{ij} > 0\} \neq \emptyset.$$ 

We solve a max-flow problem $\max_{x \in X(x^*)} x_{ts}$ on $G(N, A)$ where $x \in X(x^*)$ means $x$ satisfies the flow conservation and the following capacity constraints

$$x_{ij} \leq x^*_{ij}, \ \forall (i, j) \in A - A(x^*); x_{ij} \leq 0, \ \forall (i, j) \in A(x^*).$$ 

Since it restricts using any arcs in $A(x^*)$ in the above problem, its optimal solution, denoted as $\hat{x}(x^*)$, represents all the cycles within $x^*$ that use no arcs in $A(x^*)$. Define $\hat{x}_{ij} = x^*_{ij} - \hat{x}_{ij}(x^*)$ for any $(i, j) \in \hat{A}$. By this definition, in $\hat{x}$, any cycles passing $(t, s)$ must pass arcs in $A(x^*)$, therefore we have $\sum_{(i, j) \in A(x^*)} \hat{x}_{ij} \geq \hat{x}_{ts}$. Moreover, because $\hat{x}$ is upper bounded by $x^*$ and $x^*_{ij} \omega_{ij} = 0$ for any $(i, j) \in A - A(x^*)$, we have

$$\sum_{(i, j) \in A} \omega_{ij} \hat{x}_{ij} = \sum_{(i, j) \in A(x^*)} \hat{x}_{ij} \geq \hat{x}_{ts}, \quad (3.7)$$ 

and $\hat{x}$ is a feasible solution in $g(\omega)$, with the objective value as

$$\hat{x}_{ts} - \sum_{(i, j) \in A} \omega_{ij} \hat{x}_{ij}$$

$$= (x^*_{ts} - \sum_{(i, j) \in A} \omega_{ij} x^*_{ij}) - (\hat{x}_{ts} - \sum_{(i, j) \in A(x^*)} \omega_{ij} \hat{x}_{ij})$$

$$\geq x^*_{ts} - \sum_{(i, j) \in A} \omega_{ij} x^*_{ij},$$

where the last inequality is due to (3.7). Then by the definition of $x^*$, $\hat{x}$ is also an optimal solution to $g(\omega)$. Also notice that $\hat{x}_{ij} = 0$ for any $(i, j) \in A(x^*)$, therefore $\hat{x} \in X(\omega)$, which completes the proof.

Our method uses lower bound approximations to find the optimal solution. This bound is based on the convexity of $g(\cdot)$ and the concavity of $h(\cdot)$, which have been
shown in lemma 2.1.3 and 2.1.4.

Lemma 3.2.2 For any \( \omega \in [0, 1]^{|A|} \), there is a set of binary vectors \( \{I^i\}_{i=1}^K \), such that \( \omega \) is a convex combination of these vectors. That is,

\[
\omega = \sum_{i=1}^K w_i I^i, \quad \text{where} \quad \sum_{i=1}^K w_i = 1, \quad \text{and} \quad w_i \geq 0, \forall i = 1, \ldots, K.
\]

Proof: \([0, 1]^{|A|}\) is a unit hypercube in \(|A|\)-dimensions, with its corner points the set of binary vectors. Since the unit hypercube is a convex set polytope, therefore for any \( \omega \in [0, 1]^{|A|} \), it can be represented as a convex combination of these corner points.

Theorem 3.2.3 \( g(\omega) \leq h(\omega), \forall \omega \in [0, 1]^{|A|} \).

Proof: For any \( \omega \in [0, 1]^{|A|} \), by lemma 3.2.2, one can decompose it as a convex combination of binary vectors \( \omega = \sum_{i=1}^K w_i I^i \), where \( I^i \) is a \(|A|\)-dimension binary vector and \( w_i \geq 0, \sum_{i=1}^K w_i = 1 \). Then By theorem 3.2.1,

\[
h(I^i) = g(I^i), \quad \forall i = 1, \ldots, K.
\]

Since \( h(\omega) \) is concave and \( g(\omega) \) is convex, by Jensen's Inequality, we have

\[
g(\omega) \leq \sum_{i=1}^K w_i g(I^i) = \sum_{i=1}^K w_i h(I^i) \leq h(\omega).
\]

Theorem 3.2.1 shows that \( h(\omega) \) and \( g(\omega) \) are equal when \( \omega \) is binary, which allows us to represent the interdictor problem (3.4) as

\[
J = \min_{\gamma \in \Gamma(R)_b} \sum_{I \in \Omega} P(I) g(I \cdot \gamma) = E[g(I \cdot \gamma)]. \tag{3.8}
\]

By relaxing the feasible set from \( \Gamma(R)_b \) to \( \Gamma(R) \), because \( g(\omega) \) is convex, we can use
Jensen's inequality to get a bound on $J$

$$J \geq \min_{\gamma \in \Gamma(R)} E[I[g(I \cdot \gamma)]] \geq \min_{\gamma \in \Gamma(R)} g(E[I] \cdot \gamma)$$

(3.9)

This result (3.9) provides a lower bound on the achievable interdiction performance through the solution of a single averaged max-flow problem for each possible attack. Furthermore, the objective function is convex in the relaxed attack variables $\gamma$. Even if the bound is loose, one can improve the bound by using a partition average approach, such as the Sequence Approximation Approach in (Cormican et al., 1998).

Given a partition $\Phi$ of $\Omega$, define the binary-relaxed lower bound problem as

$$L(\Phi, \Gamma(R)) := \min_{\gamma \in \Gamma(R)} \sum_{\phi \in \Phi} P(\phi) g(\rho(\phi) \cdot \gamma),$$

(3.10)

where $P(\phi)$ is the probability of subset $\phi$ and

$$\rho(\phi)_{ij} := E[I_{ij} | I \in \phi], \forall (i, j) \in A.$$

Note when $\Phi = \{\Omega\}$, $L(\{\Omega\}, \Gamma(R)) = \min_{\gamma \in \Gamma(R)} g(E[I] \cdot \gamma)$, which is a lower bound of $J$. We will show this lower bound approximation is monotonically non-decreasing with finer partitions in Theorem 3.2.5, which is an application of the more general result in (Hausch and Ziemba, 1983).

**Lemma 3.2.4** If $\Phi^1 \leq \Phi^2$, for any $\phi^1_k \in \Phi^1$, denote $\{\phi^2_1, \ldots, \phi^2_m\} \subset \Phi^2$ such that $\phi^1_k = \cup_{i=1}^m \phi^2_i$, then

$$\rho(\phi^1_k)_{ij} = \sum_{\phi^2_i \subset \phi^1_k} \frac{P(\phi^2_i)}{P(\phi^1_k)} \rho(\phi^2_i)_{ij}, \forall (i, j) \in A.$$

**Proof:** By the definition of $\rho(\phi)$ and the conditional expectation, for any
\[(i, j) \in A\]

\[
\rho(\phi^k_{ij}) = E[I_{ij}|I_{ij} \in \phi^k_{ij}] = E[E[I_{ij}|I_{ij} \in \phi^2_m]|I_{ij} \in \phi^k_{m}] = \sum_{\phi^m \subset \phi^k} \frac{P(\phi^2_m)}{P(\phi^1_k)} \rho(\phi^2_m)_{ij}.
\]

Theorem 3.2.5 Let \(\Phi^1, \Phi^2\) be partitions of \(\Omega\) with \(\Phi^1 \leq \Phi^2\), then for any nonempty \(\Gamma\),

\[
L(\Phi^1, \Gamma) \leq L(\Phi^2, \Gamma).
\]

Proof: For any \(\gamma \in \Gamma\), the objective function in \(L(\Phi^2, \Gamma)\) is

\[
\sum_{\phi^2 \in \Phi^2} P(\phi^2)g(\rho(\phi^2) \cdot \gamma)
\]

\[
= \sum_{\phi^1 \in \Phi^1} \sum_{\phi^2 \subset \phi^1} P(\phi^2)g(\rho(\phi^2) \cdot \gamma)
\]

\[
= \sum_{\phi^1 \in \Phi^1} P(\phi^1) \sum_{\phi^2 \subset \phi^1} \frac{P(\phi^2)}{P(\phi^1)} g(\rho(\phi^2) \cdot \gamma).
\]

Because \(g(\cdot)\) is convex, by Jensen's Inequality, we have

\[
\sum_{\phi^2 \in \Phi^2} P(\phi^2)g(\rho(\phi^2) \cdot \gamma) \geq \sum_{\phi^1 \in \Phi^1} P(\phi^1)g(\sum_{\phi^2 \subset \phi^1} \frac{P(\phi^2)}{P(\phi^1)} \rho(\phi^2) \cdot \gamma)
\]

Applying lemma 3.2.4 on the right hand side, we have

\[
\sum_{\phi^2 \in \Phi^2} P(\phi^2)g(\rho(\phi^2) \cdot \gamma) \geq \sum_{\phi^1 \in \Phi^1} P(\phi^1)g(\rho(\phi^1) \cdot \gamma).
\]

Minimize \(\gamma \in \Gamma\) on both sides, then we have \(L(\Phi^1, \Gamma) \leq L(\Phi^2, \Gamma)\). \(\blacksquare\)

Theorem 3.2.5 shows that the gap between \(L(\Phi, \Gamma(R))\) and \(J\) decreases monotonically on finer partitions. We now define a particular set of partitions that will form the basis for our approximations:

Definition 3.2.6 For any subset of arcs \(D \subset A\), the partition aligned with \(D\), denoted as \(\Phi(D)\), is a partition that has \(2^{|D|}\) components, and each component \(\phi_i\) has
a single, unique outcome on the arcs in $D$ and every possible outcome on the arcs not in $D$.

For instance, if $D$ consisted of a single arc $(1, 2)$, then partition $\Phi$ would consist of two sets: $\phi_1 = \{I | I_{1,2} = 1\}$ and $\phi_2 = \{I | I_{1,2} = 0\}$. For any binary strategy $\gamma$, let the set of attacked arcs be

$$D(\gamma) := \{(i, j) | \gamma_{ij} = 1\}.$$  

Denote the expected performance of $\gamma$ as

$$u(\gamma) := \sum_{I \in \Omega} P(I) \cdot h(I \cdot \gamma)$$

and the approximation value of $\gamma$ with partition $\Phi$ as

$$l(\Phi, \gamma) := \sum_{\phi \in \Phi} P(\phi) \cdot g(\rho(\phi) \cdot \gamma),$$

we will show that the approximation value can be equal to the original value without considering all possible scenarios in $\Omega$.

**Theorem 3.2.7** If $\gamma$ is binary, then $l(\Phi(D(\gamma)), \gamma) = u(\gamma)$.

**Proof:** For any $\phi \in \Phi(D(\gamma))$ and any $I \in \phi$, if $\gamma_{ij} = 0$, then

$$\rho(\phi)_{ij} \gamma_{ij} = I_{ij} \gamma_{ij} = 0.$$

If $\gamma_{ij} = 1$, since every scenario $I$ in $\phi$ has the same outcome in arc $(i, j)$, then

$$\rho(\phi)_{ij} \gamma_{ij} = I_{ij} \gamma_{ij}.$$

As a result,

$$\rho(\phi) \cdot \gamma = I \cdot \gamma, \forall I \in \phi.$$
Note also that $P(\phi) = \sum_{I \in \phi} P(I)$. These two observations imply that

$$P(\phi)g(\rho(\phi) \cdot \gamma) = \sum_{I \in \phi} P(I)g(I \cdot \gamma) = \sum_{I \in \phi} P(I)h(I \cdot \gamma).$$

Summing over all $\phi \in \Phi(D(\gamma))$, we have $l(\Phi(D(\gamma)), \gamma) = u(\gamma)$. □

Our algorithm, Modified Branch and Bound (MBB), is based on the branch and bound method (Land and Doig, 1960), exploiting the sequence of bounds derived above and using aligned partitions that are successively refined. This approach can be illustrated as a binary tree, where, at each level of the tree, there is a decision to interdict or not to interdict a particular arc. Hence, each node consists of a feasible partial attack specified by the path down the tree, along with some free attack variables that can subsequently be optimized. The novelty of our approach is: for each node in that tree, we can define an aligned partition to the partial strategy represented in that node, and use our bounds to obtain a lower bound in performance. Thus, our bound approach has increasing resolution: early in the search, coarse partitions are used, and later in the search, finer partitions are used that lead to more accurate lower bounds, which provide guidance to select the order in which to refine nodes in the tree.

A node $k$ in the branch and bound tree consists of a subset of arcs $A_k \subset A$ with specific assigned interdiction values $\gamma_{ij} \in \{0, 1\}, \forall (i, j) \in A_k$. Let $\Gamma^k \subset \Gamma(R)$ be the feasible set of possible values for interdiction variables $\gamma_{ij}$ that are consistent with the assignments on $A_k$. Denote

$$D(\Gamma^k) := \{(i, j) | \forall \gamma \in \Gamma^k, \gamma_{ij} = 1\}$$

as the set of arcs that are surely interdicted in $\Gamma^k$, and

$$A(\Gamma^k) := D(\Gamma^k) \cup \{(i, j) | \forall \gamma \in \Gamma^k, \gamma_{ij} = 0\}$$
as the set of arcs on which actions in $\Gamma^k$ are fixed. $\Phi(D(\Gamma^k))$ denotes the partition aligned with respect to $D(\Gamma^k)$. At each node, one solves a lower bound problem

$$L(\Phi(D(\Gamma^k)), \Gamma^k) := \min_{\gamma \in \Gamma^k} \sum_{\phi \in \Phi(D(\Gamma^k))} P(\phi) \ g(\rho(\phi) \cdot \gamma) \quad (3.11)$$

and uses the optimal solution to expand the search.

However, this minimization may be non-trivial because $g(\cdot)$ is a maximization problem. We address this by using the dual of $g(\omega)$ which is a minimization problem in terms of dual variables for each node $\pi_n$ (associated with the constraint (3.3)) and for each capacitated arc $\alpha_{ij}$ (associated with the constraint (3.6))

$$Dg(\omega) := \min_{\pi, \alpha} \sum_{(i,j) \in A} u_{ij} \alpha_{ij}$$

s.t.

$$\pi_i - \pi_j \geq 1,$$

$$\alpha_{ij} + \pi_i - \pi_j \geq -\omega_{ij}, \forall (i,j) \in A$$

$$\alpha_{ij} \geq 0, \forall (i,j) \in A.$$ 

Notice that $\omega$ appears linearly as constraints in the dual formulation, not multiplying any of the dual variables. Thus, by strong duality, replacing $g(\cdot)$ with $Dg(\cdot)$, we can reformulate (3.11) as a linear program with variables $\{\gamma\}_{ij}, \{\pi\}_{i}^{\phi}, \{\alpha\}_{i}^{\phi}$

$$\min_{\gamma \in \Gamma^k} \sum_{\phi \in \Phi(D(\Gamma^k))} P(\phi) \ Dg(\rho(\phi) \cdot \gamma) \quad (3.12)$$

where there is a set of dual variables $\{\pi\}_{i}^{\phi}, \{\alpha\}_{i}^{\phi}$ for each set of outcomes $\phi$. The MBB algorithm is outlined in Algorithm 4. Typically, the number of sets in the aligned partition $|\Phi(D(\Gamma^k))|$ is much smaller than the number of outcomes $|\Omega|$, thus the number of terms required to evaluate a bound of $u(\gamma^k)$ is much smaller than the number of possible outcomes. The resulting algorithm uses a loose bound based on a coarse partition early in the process, and uses an increasingly accurate bound as the
Procedure 4 Branch and Bound Method for SNIP
1: Initialize $\gamma^* \leftarrow 0, J^* \leftarrow \text{max-flow without attack.}$
2: Solve $L([\Omega], \Gamma(R))$, denote the optimal value $L^0$ and solution $\gamma^0$.
3: Initialize the set of candidate branch(es) $Q \leftarrow \{(\gamma^0, L^0, [\Omega])\}$.
4: while $Q \neq \emptyset$ do
5: Select and remove a node $(\gamma^k, L^k, \Gamma^k)$ from $Q$.
6: if $\gamma^k$ is binary then
7: Evaluate $u(\gamma^k) = \sum_{\phi \in \Phi(D(\gamma^k))} P(\phi) g(\rho(\phi) \cdot \gamma^k)$.
8: if $u(\gamma^k) < J^*$ then
9: Update $\gamma^* \leftarrow \gamma^k, J^* \leftarrow u(\gamma^k)$.
10: Remove any node $(\tilde{\gamma}, \tilde{L}, \tilde{\Gamma})$ in $Q$ if $\tilde{L} \geq J^*$.
11: end if
12: end if
13: Select an arc that have the maximum fraction interdiction $\tau_{ij}^k$,

$$(i^*, j^*) \in \arg \max_{(i,j) \in A(\Gamma^k)} \gamma_{ij}^k,$$

14: Split $\Gamma^k$ on $(i^*, j^*)$ such that,
15: $\Gamma_{k,0} := \Gamma^k \cap \{r_{i^*,j^*} = 0\}$ and $\Gamma_{k,1} := \Gamma^k \cap \{r_{i^*,j^*} = 1\}$.
16: for $c = 0, 1$ do
17: Solve $L(\Phi(D(\Gamma_{k,c})), \Gamma_{k,c})$, denote the optimal value $L_{k,c}^c$ and solution $\gamma_{k,c}^c$.
18: if $L_{k,c}^c < J^*$ then
19: add $(\gamma_{k,c}^c, L_{k,c}^c, \Gamma_{k,c}^c)$ to $Q$.
20: end if
21: end for
22: end while
23: Output $\gamma^*, J^*$ as the optimal solution and value.
partitions are refined deeper in the branch and bound tree.

**Implementations of Subroutines**  There are two important subroutines in MBB: solving the lower bound problem \( L(\Phi(\Gamma^k), \Gamma^k) \) and selecting the next candidate node \( \Gamma^k \) in the branch and bound tree to explore.

Notice \( L(\Phi(\Gamma^k), \Gamma^k) \) has the stochastic programming structure with a specified subset of scenarios, we implement the \( L\)-Shape decomposition method (Section 2.2), which is widely used to solve stochastic programming problems. We have two implementations whose difference is on the master problems. The original L-Shaped method (Decomposition 1) has the master (2.12) and the alternative decomposition method (Decomposition 2) has the master problem (2.14). We will compare the performance of these two methods with numerical examples.

Another subroutine consists of selecting the order in which nodes in the set \( Q \) are explored. Several criteria are explored here.

- "Best First Search" (BFS): select the best branch, which has the lowest \( L^k \).
- "Depth First Search" (DFS): select the deepest branch, which has the largest \( |A(\Gamma^k)| \).

BFS and DFS are two traditional branching criteria in Branch and bound method. BFS is often superior to DFS, especially when the optimal solution has shallow depth and the objective function is smooth (neighboring solutions have similar objective values).

- "Distributed Best First Search" (DBFS): select the best branch from top, then choose the best one among those with one more depth. After reaching the bottom, return to the top and repeat this process.
• "Least Depth First Search" (LDFS): similar to (DBFS) but instead of choosing the next within the candidates with one more depth, it chooses one with AT LEAST one more depth.

DBFS is a criteria introduced by Kao et al. (G.K.Kao et al., 2009), which is a hybrid between DFS and BFS. By combining DFS and BFS, DBFS is designed to find an optimal solution earlier than DFS. With similar intuition, we extend it to LDFS, which reaches the bottom more quickly than DBFS. Notice in our problem, the number of surely interdicted arcs $|D(\Gamma^k)|$ (called score) should be a more proper to measure the closeness to the global solution than $|A(\Gamma^k)|$. Therefore, replacing the depth with $|D(\Gamma^k)|$ in DFS, DBFS, LDFS respectively, we get

• "Score First Search" (SFS): replacing $|A(\Gamma^k)|$ with $|D(\Gamma^k)|$ in DFS.

• "Distributed Score First Search" (DSFS): replacing $|A(\Gamma^k)|$ with $|D(\Gamma^k)|$ in DBFS.

• "Least Score First Search" (LSFS): replacing $|A(\Gamma^k)|$ with $|D(\Gamma^k)|$ in LDFS.

**Compared with Sequential Approximation Algorithm**  Both our method and Sequential Approximation Algorithm (SAA) (see Section 2.1) utilize the idea of improving bounds sequentially by more refined partitions. However, our method differs from their method in the following aspects.

First of all, in each iteration, SAA solves the problem of $LBMIN(\Phi)$, which is a mixed integer linear programming (MILP) problem with a constant feasible set. In our method, our lower bounds are obtained by solving a LP problem $L(\Phi, \Gamma)$ with shrinking feasible set. We get a fractional solution, rather than a binary solution in SAA. We depend on the branch and bound using these lower bounds to obtain an integer solution later.
Furthermore, SAA's partition procedure requires lots of computations to select a cell and an arc within the cell to split. To select a cell, SAA calculates the gaps between the upper and lower bounds for all cells. After a cell is selected, SAA also needs to check all possible splits, each of which requires additional computation of solving 4 optimization problems. In contrast, we partition based on the fractional solution of \( L(\Phi, \Gamma) \), which does not need any computation to generate the new partition for next iteration.

Finally, the implementation of SAA requires the availability of a tight upper bound problem (e.g. \( h(\omega) \)) because it needs to calculate the gap between bounds for cells and for splitting arcs. However our method does not need this. Therefore our method can be extended to solve problems that do not have a tight upper bound. In fact, we extend our method to solve the multi-stage interdiction problem studied in chapter 5.

Fig. 3.2 illustrates how MBB solves an interdiction problem on a network with 5 attackable arcs and the attacker has budget to attack at most 3 arcs. MBB starts at the root node where the initial partition is \( \Phi = \{\Omega\} \). By the solution to the

Figure 3.2: MBB solving an interdiction problem on a network with 5 attackable arcs where the attacker can attack at most 3 arcs.
approximation problem of $L(\Phi, \Omega(R))$, MBB splits on arc 2 according. Then the feasible set of $\gamma$ has been separated into two directions: one surely attack arc 2, $\Gamma_1 := \{\gamma|\gamma_2 = 1\}$ and the other surely not to attack arc 2, $\Gamma_0 := \{\gamma|\gamma_2 = 0\}$. Correspondingly, MBB aligns the partition of outcome space with the split feasible sets: $\Phi_1 := \{\{I_1I_2 = 1\}, \{I_1I_2 = 0\}\}$ for $\Gamma_1 := \{\gamma|\gamma_2 = 1\}$ and $\Phi = \{\Omega\}$ for $\Gamma_1 := \{\gamma|\gamma_2 = 0\}$. After another split on $\Gamma_0 := \{\gamma|\gamma_2 = 0\}$, MBB reaches node $A$ where the lower bound problem outputs a binary solution. Since the exact value of this solution is better than current optimal value, we update the optimal solution at this node. Similarly we update optimal solution at node $B, C$ since we get better binary solutions there. At node $D, E$, MBB has binary solutions, however, they are not better than current optimal solution. Therefore we discard them. At the two $X$ nodes, MBB discards them because the lower bound values at these nodes are greater than current optimal value. At last, MBB outputs the optimal solution at node $C$, i.e., attack arcs 1, 4, 5. As one can see, the largest partition size is $2^3 = 8$, much less than the total scenario number $2^5 = 32$.

3.3 Numerical Results

In this section we compare our method (MBB) with other two methods, Sequential Approximation Algorithm (SAA) (Cormican et al., 1998) and Sample Averaging Approach (SAM) (Janjarassuk and Linderoth, 2008), on simulated network interdiction problems, which were also used as examples in both papers. To highlight the efficiency of those methods, we also include the performance of an enumeration method (ENU), which enumerates only the feasible strategies on the border of the budget constraint, thus limits the number of strategies to evaluate. ENU can be viewed as a brute force benchmark to illustrate the difficulties of solving these problems. All methods are coded in C with CPLEX Callable Library (version 12.0) and all nu-
Figure 3-3: SNIP 7 × 5: 37 nodes and 72 arcs, 22 of which are interdictable.

Numerical experiments in this paper are run on a 64bit Windows 7 machine with Intel i5 - 2430 CPU (Dual core, 2.4 GHz) and 4 GB memory.

The underlying networks are shown in Fig. 3-3 (SNIP 7 × 5) and Fig. 3-4 (SNIP 4 × 9). The capacities of arcs are shown on the graph. If the capacity has a tilde on top, then \( p_{ij} = 0.75 \), the probability of successful interdiction; otherwise, \( p_{ij} = 0 \) and that arc cannot be interdicted. The resource required to interdict each arc is \( c_{ij} = 1, \forall (i, j) \in A \). The interdiction budget \( R \) ranges from 5 to 9. When \( R \) increases, the number of feasible strategies increases exponentially, which in turn raises the difficulty of solving that problem. Simply, we can take \( R \) as an indicator of the hardness of problems.

Notice that MBB have two important subroutines: solving LP problems and selecting candidate nodes in the branch and bound tree. We explore the combinations
of them to find the best one. Table 3.1 shows the running time of solving the interdiction problem on SNIP $7 \times 5$ (Budget = 6) with different combinations of subroutines. “Sub#” is the number of calling the network solver for $g(\omega)$, which is a relative measure of computation since the network solver is frequently called and it takes up most of time in our method. The results show that the choice of decomposition method does not affect the processing time. The branching strategy also has little impact in the overall performance, although there is a preference for avoiding simple algorithms such as depth-first search. This may due to the example is too small to generate a deep Branch and Bound tree. Without specific instruction, in the rest of this paper, our method uses “Decomposition 1” and “BFS”.

We compare MBB with SAA and ENU in Tab. 3.2, which shows the running times in solving interdiction problems on the above two networks. MBB performs best in all instances and is about 3 orders of magnitude faster than the benchmark ENU. Compared with SAA, MBB is faster in about 2 orders of magnitude. For instance, in SNIP $7 \times 5$ with budget 5, SAA spends about 3 second while MBB runs 0.06. Its efficiency becomes more obvious when the problem is harder. For SNIP $4 \times 9$ with budget 9, SAA needs 1127.06 seconds, compared with 2.87 seconds in MBB.

Next we compare MBB with SAM (see Section 2.3), which is a random algorith-
Table 3.1: Performance of Different Branching Criteria and Decomposition Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Decomposition 1</th>
<th>Decomposition 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time(s)</td>
<td>sub#</td>
</tr>
<tr>
<td>BFS</td>
<td>0.0880</td>
<td>3162</td>
</tr>
<tr>
<td>DFS</td>
<td>0.0850</td>
<td>2837</td>
</tr>
<tr>
<td>DBFS</td>
<td>0.0930</td>
<td>2960</td>
</tr>
<tr>
<td>DSFS</td>
<td>0.0830</td>
<td>2694</td>
</tr>
<tr>
<td>SFS</td>
<td>0.0860</td>
<td>2799</td>
</tr>
<tr>
<td>LSFS</td>
<td>0.0830</td>
<td>2684</td>
</tr>
<tr>
<td>LDFS</td>
<td>0.0910</td>
<td>2980</td>
</tr>
</tbody>
</table>

Table 3.2: Compare MBB with SAA on solving SNIPs

<table>
<thead>
<tr>
<th>Budget</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SNIP 7 × 5, second</td>
<td></td>
<td>SNIP 4 × 9, second</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENU</td>
<td>5.34</td>
<td>28.44</td>
<td>127.57</td>
<td>483.20</td>
<td>1499.55</td>
</tr>
<tr>
<td>MBB</td>
<td>0.06</td>
<td>0.09</td>
<td>0.17</td>
<td>0.25</td>
<td>0.58</td>
</tr>
<tr>
<td>SAA</td>
<td>2.45</td>
<td>5.26</td>
<td>6.12</td>
<td>9.06</td>
<td>17.74</td>
</tr>
<tr>
<td></td>
<td>SNIP 7 × 5, second</td>
<td></td>
<td>SNIP 4 × 9, second</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENU</td>
<td>11.73</td>
<td>58.30</td>
<td>290.02</td>
<td>1312.87</td>
<td>4846.50</td>
</tr>
<tr>
<td>MBB</td>
<td>0.03</td>
<td>0.05</td>
<td>0.25</td>
<td>0.72</td>
<td>2.87</td>
</tr>
<tr>
<td>SAA</td>
<td>1.97</td>
<td>11.32</td>
<td>60.20</td>
<td>112.12</td>
<td>1127.06</td>
</tr>
</tbody>
</table>
m that uses Monte Carlo technique to solve SNIPs approximately. SAM solves $M$
approximation problems, each considers $N$ independent samples. SAM uses $E$
independent samples to evaluate the expected values of these solutions and then selects
the one with the smallest value. In (Janjarassuk and Linderoth, 2008), they imple-
mented SAM with $M = 10$, $E = 10^5$ and $N$ ranges from 50 to 5000. We adapt the
same set of parameters in our implementations. In SAM1, $(N, M, E) = (50, 10, 10^5)$
and in SAM2, $(N, M, E) = (5000, 10, 10^5)$.

Tab. 3.3 shows the running times of these methods. Notice SAM is a random
algorithm, which should be accessed by its average performance. We ran SAM 100
times to solve each problem and provided average performance in Tab. 3.3. We also
estimate the output solutions by relative error (defined as the solution’s objective
value over the optimal value and minus one) and its standard deviation.

MBB is much faster than SAM in all cases, in about 2 orders of magnitude. This is
more obvious in small problems (with small $R$) due to the computation increment with
$R$ is slower in SAM than in MBB. SAM’s computation is affected by the subproblem’s
size. When the size increases from 50 to 5000, its computation is almost doubled.
However, additional computation can improve the output’s quality. For example, in
the case of SNIP $4 \times 9$ with $R = 9$, the average error of SAM1 is $0.15\%$ with standard
deviation of $0.31\%$. After increasing subproblem’s size $N$ from 50 to 5000, the error
decreases to $0.01\%$ with standard deviation of $0.01\%$.

As the numerical results indicate, MBB performs much better than the other two
methods. It’s about two orders of magnitude faster than SAA. When compared with
SAM, it runs faster and its outputs are guaranteed to be optimal.
### Table 3.3: Compare MBB with SAM on solving SNIPs

<table>
<thead>
<tr>
<th>Budget</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENU</td>
<td>5.34</td>
<td>28.44</td>
<td>127.57</td>
<td>483.20</td>
<td>1499.55</td>
</tr>
<tr>
<td>MBB</td>
<td>0.06</td>
<td>0.09</td>
<td>0.17</td>
<td>0.25</td>
<td>0.58</td>
</tr>
<tr>
<td>SAM1</td>
<td>8.24</td>
<td>8.65</td>
<td>9.33</td>
<td>10.40</td>
<td>14.39</td>
</tr>
<tr>
<td>err</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>std</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>SAM2</td>
<td>15.26</td>
<td>17.97</td>
<td>19.01</td>
<td>24.45</td>
<td>35.50</td>
</tr>
<tr>
<td>err</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>std</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

**SNIP 7 × 5, second**

| ENU    | 11.73| 58.30| 290.02| 1312.87| 4846.50|
| MBB    | 0.03| 0.05| 0.25| 0.72| 2.87|
| SAM1   | 10.18| 12.96| 14.62| 17.20| 19.61|
| err    | 0%  | 0.07%| 0.05%| 0.07%| 0.16%|
| std    | 0%  | 0.12%| 0.08%| 0.53%| 0.31%|
| SAM2   | 15.40| 23.65| 47.06| 52.91| 95.23|
| err    | 0%  | 0.02%| 0.04%| 0.01%| 0.01%|
| std    | 0%  | 0.07%| 0.04%| 0.01%| 0.01%|

**SNIP 4 × 9, second**
Chapter 4

Extended Stochastic Network Interdiction Problems

The basic model in chapter 3 considers a directed network with a single source and a single destination. In some applications, the network is undirected. It may have multiple sources/destinations, or the existences of these sources/destinations are uncertain. In this chapter we will extend the model in the previous chapter to incorporate these features and extend our algorithms to solve these models. We first discuss these extensions and formulate corresponding bounds separately. Then in section 4.4, we develop solutions for these models. Finally we implement our algorithms to solve interdiction problems on a real power grid, IEEE BUS 300 System, which is modeled as an undirected network with multiple sources/destinations.

4.1 Interdiction of Undirected Networks

Given an undirected network $G(N, A^U)$ where the arcs in $A^U$ are denoted as $(i, j)$ with $i < j$, one can convert it into an equivalent directed network by doubling each undirected arc into two directed arcs with opposite directions, both of which inherit the capacity constraint of that undirected arc. Denote the set of the directed arcs as

$$A := \{(i, j) \mid (i, j) \text{ or } (j, i) \in A^U\}.$$
Then we have a directed network $G(N, A)$. By introducing an artificial arc $(t, s)$ from $t$ to $s$ with infinite capacity and denote the new set of arcs as $\overline{A}$, the flow conservation constraint is

$$
\sum_{(n,i)\in \overline{A}} x_{ni} - \sum_{(j,n)\in \overline{A}} x_{jn} = 0, \forall n \in N.
$$

(4.1)

The conditional capacity constraint is different from the basic model. Let $\omega_{ij}, \forall (i, j) \in A^U$ be the availability of arcs in $A^U$, each of which corresponds to two arcs in $A$. Denote the capacity of arcs in $A^U$ as $u$, then given $\omega$, for any $(i, j) \in A$

$$
x_{ij} \leq u_{ij}(1 - \omega_{ij}), \text{ if } i < j; \quad x_{ij} \leq u_{ji}(1 - \omega_{ji}), \text{ otherwise.}
$$

(4.2)

The max-flow problem is

$$
h^U(\omega) := \max_{x \in X^U(\omega)} x_{ts},
$$

(4.3)

where $x \in X^U(\omega)$ means $x \geq 0$ and satisfies both constraints in (4.1) and (4.2). Define the penalty problem as

$$
g^U(\omega) := \max_{x \in X^U} x_{ts} - \sum_{(i,j) \in A^U} \omega_{ij}(x_{ij} + x_{ji}),
$$

(4.4)

where $x \in X^U$ means $x \geq 0$, satisfies (4.1) and for any $(i, j) \in A$

$$
x_{ij} \leq u_{ij}, \text{ if } i < j; \quad x_{ij} \leq u_{ji}, \text{ otherwise.}
$$

(4.5)

**Theorem 4.1.1.** $h^U(\omega) = g^U(\omega)$, for any $\omega \in \{0, 1\}^{|A|}$.

**Proof:** For any $x \in X^U(\omega)$, for any $(i, j) \in A^U$ it's either $\omega_{ij} = 0$ or $x_{ij} = x_{ij} = 0$ due to capacity constraint, then we have $x_{ts} - \sum_{(i,j)\in A^U} \omega_{ij}(x_{ij} + x_{ji}) = x_{ts}$. Therefore when $x \in X^U(\omega)$, problems $h^U(\omega)$ and $g^U(\omega)$ have the same objective value. Notice $X^U(\omega) \subset X^U$, now we just need to show that $g^U(\omega)$ always has an optimal solution in $X^U(\omega)$. 
Let $x^*$ be an optimal solution to $g^U(\omega)$, such that $x^* \notin X^U(\omega)$, i.e.

$$A(x^*) := \{(i, j) | (i, j) \in A, \omega_{ij} = 1 \text{ or } \omega_{ji} = 1; \text{ and } x^*_{ij} > 0\} \neq \emptyset.$$ 

We solve a max-flow problem $\max_{x \in X(x^*)} x_{ts}$ on $G(N, A)$ where $x \in X(x^*)$ means $x$ satisfies the flow conservation and the following capacity constraints

$$x_{ij} \leq x^*_{ij}, \forall (i, j) \in A \setminus A(x^*) ; x_{ij} \leq 0, \forall (i, j) \in A(x^*).$$ 

Since it restricts using any arcs in $A(x^*)$ in the above problem, its optimal solution, denoted as $\tilde{x}(x^*)$, represents the cycles within $x^*$ that use no arcs in $A(x^*)$. Define $\tilde{x}_{ij} = x^*_{ij} - \tilde{x}_{ij}(x^*)$ for any $(i, j) \in \overline{A}$, by this definition, in $\tilde{x}$, any cycles passing $(t, s)$ must pass arcs in $A(x^*)$, therefore we have $\sum_{(i, j) \in A(x^*)} \tilde{x}_{ij} \geq \tilde{x}_{ts}$. Moreover, because $\tilde{x}$ is upper bounded by $x^*$ and $x^*_{ij} \omega_{ij} = 0$ for any $(i, j) \in A \setminus A(x^*)$, we have

$$\sum_{(i, j) \in A^U} \omega_{ij}(\tilde{x}_{ij} + \tilde{x}_{ji}) = \sum_{(i, j) \in A(x^*)} \tilde{x}_{ij} \geq \tilde{x}_{ts}, \quad (4.6)$$

and $\tilde{x}$ is a feasible solution in $g^U(\omega)$, with the objective value as

$$\tilde{x}_{ts} - \sum_{(i, j) \in A^U} \omega_{ij}(\tilde{x}_{ij} + \tilde{x}_{ji})$$

$$= [x^*_{ts} - \sum_{(i, j) \in A^U} \omega_{ij}(x^*_{ij} + x^*_{ji})] - [\tilde{x}_{ts} - \sum_{(i, j) \in A^U} \omega_{ij}(\tilde{x}_{ij} + \tilde{x}_{ji})]$$

$$\geq x^*_{ts} - \sum_{(i, j) \in A^U} \omega_{ij}(x^*_{ij} + x^*_{ji}),$$

where the last inequality is due to (4.6). Then by the definition of $x^*$, $\tilde{x}$ is also an optimal solution to $g^U(\omega)$. Also notice that $\tilde{x}_{ij} = 0$ for any $(i, j) \in A(x^*)$, therefore $\tilde{x} \in X(\omega)$, which completes the proof.

Next we show the convexity of $g^U(\omega)$ and the concavity of $h^U(\omega)$ after relaxing $\omega$ to be fractional.
Lemma 4.1.2 $h^U(\omega)$ is concave on $\omega \in [0, 1]^{|A^U|}$.

Proof: For any $\omega^1, \omega^2 \in [0, 1]^{|A^U|}$, let $x^s$ be the optimal solution to $h^U(\omega^s), s = 1, 2$, then we have

$$\lambda x^1_{ts} + (1 - \lambda)x^2_{ts} = \lambda h^U(\omega^1) + (1 - \lambda)h^U(\omega^2), \forall \lambda \in [0, 1].$$

Since for any $(i, j) \in A^U$, $x^k_{ij}, x^k_{ji} \leq u_{ij}(1 - \omega^k_{ij}), k = 1, 2$, then

$$\lambda x^1_{ij} + (1 - \lambda)x^2_{ij}, \lambda x^1_{ji} + (1 - \lambda)x^2_{ji} \leq u_{ij}[1 - (\lambda \omega^1_{ij} + (1 - \lambda)\omega^2_{ij})],$$

which is the capacity constraint in $h^U(\lambda \omega^1 + (1 - \lambda)\omega^2)$. Moreover, $\lambda x^1_{ij} + (1 - \lambda)x^2_{ij}$ satisfies flow conservation since $x^k, k = 1, 2$ satisfy flow conservation, then $\lambda x^1 + (1 - \lambda)x^2$ is a feasible solution in $h^U(\lambda \omega^1 + (1 - \lambda)\omega^2)$. Furthermore, since $h^U(\omega)$ is maximizing over $x$, therefore

$$h^U(\lambda \omega^1 + (1 - \lambda)\omega^2) \geq \lambda x^1_{ts} + (1 - \lambda)x^2_{ts} = \lambda h^U(\omega^1) + (1 - \lambda)h^U(\omega^2).$$

So $h^U(\omega)$ is concave on $\omega \in [0, 1]^{|A^U|}$. \qed

Lemma 4.1.3 $g^U(\omega)$ is convex on $\omega \in [0, 1]^{|A^U|}$. 
Proof: For any $\omega^1, \omega^2 \in [0, 1]^{A^U}$, $\lambda \in [0, 1]$, let $g^U(\lambda \omega^1 + (1 - \lambda)\omega^2)$'s solution to be $\bar{x}$, then

$$g^U(\lambda \omega^1 + (1 - \lambda)\omega^2)$$

$$= \max_{x \in X} x_{ts} - \sum_{(i,j) \in A^U} (\lambda \omega^1_{ij} + (1 - \lambda)\omega^2_{ij})(x_{ij} + x_{ji})$$

$$= \bar{x}_{ts} - \sum_{(i,j) \in A^U} (\lambda \omega^1_{ij} + (1 - \lambda)\omega^2_{ij})(\bar{x}_{ij} + \bar{x}_{ji})$$

$$= \lambda [\bar{x}_{ts} - \sum_{(i,j) \in A^U} \omega^1_{ij}(\bar{x}_{ij} + \bar{x}_{ji})] + (1 - \lambda)[\bar{x}_{ts} - \sum_{(i,j) \in A^U} \omega^2_{ij}(\bar{x}_{ij} + \bar{x}_{ji})]$$

$$\leq \lambda g^U(\omega^1) + (1 - \lambda)g^U(\omega^2)$$

The last inequality is because $g^U(\omega)$ maximizes on $\omega$ and $\bar{x}$ is feasible in both $g^U(\omega^1)$ and $g^U(\omega^2)$.

Theorem 4.1.4 $g^U(\omega) \leq h^U(\omega)$, $\forall \omega \in [0, 1]^{A^U}$.

Proof: The proof is similar to Theorem 3.2.3, which utilizes the convexities of $g^U(\omega), h^U(\omega)$ with Jensen's Inequality.

As a summary, $h^U(\omega)$ and $g^U(\omega)$ satisfy

P1 $\forall \omega \in \{0, 1\}^{A^|}, h^U(\omega) = g^U(\omega)$.

P2 $\forall \omega \in [0, 1]^{A^|}, h^U(\omega)$ is concave and $g^U(\omega)$ is convex.

P3 $\forall \omega \in [0, 1]^{A^|}, g^U(\omega) \leq h^U(\omega)$.

Now we can develop lower bounds as before, and integrate them in a branch and bound context. We will show that in details in Section 4.4.

4.2 Interdiction of Multi-source/destination Network

Models with multi-source/destination networks are more general and they are important in real world applications. For example, in the smuggling network, there may be
more than one smuggler. In the military deployment, the enemy probably has more than one supply. Therefore we extend the model to cover multi-source/destination networks.

Some previous works also consider interdictions on multi-source/destination networks. Akgün et al. (Akgün et al., 2011) studied a deterministic interdiction problem with undirected network, which has multiple sources/destinations. The attacker minimizes flows from the sources and to the destinations. The formulation in this section can be viewed as an extension of Akgün's model. Compared to Akgün's model, our model considers stochastic problems rather than deterministic problems. Furthermore, our objective function is more flexible, allowing the attacker to weight flows differently based on sources/destinations.

Consider a directed network $G(N, A)$ with nodes $N$ and arcs $A$ that transports homogeneous goods from $K$ sources to $L$ destinations, which are denoted as $s_k, t_l \in N$ with $k = 1, \ldots, K; l = 1, \ldots, L$, respectively. Let $u_{ij}$ be the capacity of arc $(i, j)$, which is bounded. Assume that any source cannot be a destination and vice versa, i.e. for any $s_k, t_l, s_k \neq t_l$, therefore there is an implicit cap on the max-flow of the network. The network has different weights on flows based on their sources/destinations. Let $c_{s_k}$ (or $c_{t_l}$) denotes the unit price for flows from $s_k$ (or to $t_l$).

Extend the network with a virtual node $v$ and virtual arcs $(t_l, v), (v, s_k)$ for any $s_k, t_l$ with infinite capacities. Let $\overline{A}, \overline{N}$ be the set of arcs and the set of nodes in the extended network. Given a network state $\omega$, the network maximizes

$$ h^H(\omega) := \max_{x \in X^H(\omega)} \sum_{k=1}^{K} c_{s_k} x_{v,s_k} + \sum_{l=1}^{L} c_{t_l} x_{t_l,v} \quad (4.7) $$

where $x \in X^H(\omega)$ means $x \geq 0$ and satisfies the conditional capacity constraints

$$ x_{ij} \leq u_{ij}(1 - \omega_{ij}), \forall (i, j) \in A \quad (4.8) $$
and the flow conservation constraints
\[
\sum_{(n,i) \in A} x_{ni} - \sum_{(j,n) \in A} x_{jn} = 0, \forall n \in \bar{N}.
\] (4.9)

Define the penalty problem as
\[
g^H(\omega) := \max_{x \in X^H} \sum_{k=1}^{K} c_{sk} x_{v,s_k} + \sum_{l=1}^{L} c_{tl} t_{tl,v} - c_H \sum_{(i,j) \in A} \omega_{ij} x_{ij}.
\] (4.10)

where \( c_H > \max_k c_{sk} + \max_l c_{tl} \); \( x \in X^H \) means \( x \geq 0 \), satisfies the flow conservation constraint (4.9) and capacity constraints
\[
x_{ij} \leq u_{ij}, \forall (i, j) \in A
\] (4.11)

**Theorem 4.2.1** \( h^H(\omega) = g^H(\omega) \), for any \( \omega \in \{0, 1\}^{|A|} \).

**Proof:** For any \( x \in X(\omega) \), for any \((i, j)\) it's either \( \omega_{ij} = 0 \) or \( x_{ij} = 0 \) due to capacity constraint, then we have \( c_H \sum_{(i,j) \in A} \omega_{ij} x_{ij} = 0 \). Therefore when \( x \in X^H(\omega) \), problems \( h^H(\omega) \) and \( g^H(\omega) \) have the same objective value. Notice \( X^H(\omega) \subset X^H \), now we just need to show that \( g^H(\omega) \) always has an optimal solution in \( X^H(\omega) \).

Let \( x^* \) be an optimal solution to \( g^H(\omega) \), such that \( x^* \not\in X^H(\omega) \), i.e.
\[
A(x^*) := \{(i, j) | (i, j) \in A, \omega_{ij} = 1 \text{ and } x^*_{ij} > 0\} \neq \emptyset.
\]

We solve a problem on \( G(N, A) \)
\[
\max_{x \in X(x^*)} c_H \sum_{(i,j) \in A(x^*)} x_{ij} - \sum_{k=1}^{K} c_{sk} x_{v,s_k} + \sum_{l=1}^{L} c_{tl} t_{tl,v},
\]

where \( x \in X(x^*) \) means \( x \) satisfies the flow conservation and the following capacity constraints
\[
x_{ij} \leq x^*_{ij}, \forall (i, j) \in A - A(x^*); x_{ij} \leq 0, \forall (i, j) \in A(x^*).
\]
Since it restricts using any arcs in $A(x^*)$ in the above problem, its optimal solution, denoted as $\tilde{x}(x^*)$, represents the cycles within $x^*$ that use no arcs in $A(x^*)$. Define $\tilde{x}_{ij} = x_{ij}^* - x_{ij}(x^*)$ for any $(i, j) \in \bar{A}$, by this definition, in $\tilde{x}$, any cycles passing $v$ must pass arcs in $A(x^*)$, then

$$
\sum_{(i,j)\in A(x^*)} \tilde{x}_{ij} \geq \sum_{k=1}^{K} \tilde{x}_{v,s_k} + \sum_{l=1}^{L} \tilde{x}_{t_l,v}.
$$

Furthermore, because $c_H > \max_k c_{s_k} + \max_l c_{t_l}$, we have

$$
c_H \sum_{(i,j)\in A(x^*)} \tilde{x}_{ij} \geq \sum_{k=1}^{K} c_{s_k} \tilde{x}_{v,s_k} + \sum_{l=1}^{L} c_{t_l} \tilde{x}_{t_l,v}.
$$

Moreover, because $\tilde{x}$ is upper bounded by $x^*$ and $x_{ij}^* \omega_{ij} = 0$ for any $(i, j) \in A - A(x^*)$, we have

$$
c_H \sum_{(i,j)\in A} \omega_{ij} \tilde{x}_{ij} = c_H \sum_{(i,j)\in A(x^*)} \tilde{x}_{ij} \geq \sum_{k=1}^{K} c_{s_k} \tilde{x}_{v,s_k} + \sum_{l=1}^{L} c_{t_l} \tilde{x}_{t_l,v},
$$

and $\tilde{x}$ is a feasible solution in $g^H(\omega)$, with the objective value as

$$
\sum_{k=1}^{K} c_{s_k} \tilde{x}_{v,s_k} + \sum_{l=1}^{L} c_{t_l} \tilde{x}_{t_l,v} - c_H \sum_{(i,j)\in A} \omega_{ij} \tilde{x}_{ij}
$$

where the last inequality is due to (4.12). Then by the definition of $x^*$, $\tilde{x}$ is also an optimal solution to $g^H(\omega)$. Also notice that $\tilde{x}_{ij} = 0$ for any $(i, j) \in A(x^*)$, therefore $\tilde{x} \in X(\omega)$, which completes the proof.

Similar to $h(\omega)$ and $g(\omega)$, $h^H(\omega)$ is concave and $g^H(\omega)$ is convex, which will be
shown in the following lemmas.

**Lemma 4.2.2** $h^H(\omega)$ is concave on $\omega \in [0, 1]^{|A|}$.

*Proof:* For any $\omega^1, \omega^2 \in [0, 1]^{|A|}$, let $x^s$ be an optimal solution to $h^H(\omega^s), s = 1, 2$, then for any $\lambda$ in $[0, 1]$, we have

$$
\lambda \left[ \sum_{k=1}^{K} c_{sk} x_{v,s,k}^1 + \sum_{l=1}^{L} c_{l,t} x_{t,l,v}^1 \right] + (1-\lambda) \left[ \sum_{k=1}^{K} c_{sk} x_{v,s,k}^2 + \sum_{l=2}^{L} c_{l,t} x_{t,l,v}^2 \right] = \lambda h^H(\omega^1) + (1-\lambda) h^H(\omega^2).
$$

Since for any $(i, j) \in A$, $x_{ij}^k \leq u_{ij} (1 - \omega^k_{ij}), k = 1, 2$, then

$$
\lambda x_{ij}^1 + (1-\lambda) x_{ij}^2 \leq u_{ij} \left[ 1 - (\lambda \omega_{ij}^1 + (1-\lambda) \omega_{ij}^2) \right],
$$

which is the capacity constraint in $h^H(\lambda \omega^1 + (1-\lambda) \omega^2)$. Moreover, $\lambda x_{ij}^1 + (1-\lambda) x_{ij}^2$ satisfies flow conservation since $x^k, k = 1, 2$ satisfy flow conservation, then $\lambda x^1 + (1-\lambda) x^2$ is a feasible solution in $h^H(\lambda \omega^1 + (1-\lambda) \omega^2)$. Furthermore, since $h^H(\omega)$ is maximizing over $x$, therefore

$$
h^H(\lambda \omega^1 + (1-\lambda) \omega^2) \\
\geq \lambda \left[ \sum_{k=1}^{K} c_{sk} x_{v,s,k}^1 + \sum_{l=1}^{L} c_{l,t} x_{t,l,v}^1 \right] + (1-\lambda) \left[ \sum_{k=1}^{K} c_{sk} x_{v,s,k}^2 + \sum_{l=2}^{L} c_{l,t} x_{t,l,v}^2 \right] \\
= \lambda h(\omega^1) + (1-\lambda) h(\omega^2).
$$

So $h^H(\omega)$ is concave on $\omega \in [0, 1]^{|A|}$. □

**Lemma 4.2.3** $g^H(\omega)$ is convex on $\omega \in [0, 1]^{|A|}$.

*Proof:* For any $\omega^1, \omega^2 \in [0, 1]^{|A|}, \lambda \in [0, 1]$, let $\tilde{x}$ be an optimal solution to
\[ g^H(\lambda \omega^1 + (1 - \lambda)\omega^2), \text{ then} \]

\[
g^H(\lambda \omega^1 + (1 - \lambda)\omega^2) = \max_{x \in X} \sum_{k=1}^{K} c_{s_k} x_{v,s_k} + \sum_{l=2}^{L} c_t i_{t_l,v} - c_H \sum_{(i,j) \in A} (\lambda\omega_{ij}^1 + (1 - \lambda)\omega_{ij}^2) x_{ij}
\]

\[
= \sum_{k=1}^{K} c_{s_k} \tilde{x}_{v,s_k} + \sum_{l=2}^{L} c_t \tilde{x}_{t_l,v} - c_H \sum_{(i,j) \in A} (\lambda\omega_{ij}^1 + (1 - \lambda)\omega_{ij}^2) \tilde{x}_{ij} \]

\[
= \lambda \left( \sum_{k=1}^{K} c_{s_k} \tilde{x}_{v,s_k} + \sum_{l=2}^{L} c_t \tilde{x}_{t_l,v} - c_H \sum_{(i,j) \in A} \omega_{ij}^1 \tilde{x}_{ij} \right) \\
+ (1 - \lambda) \left( \sum_{k=1}^{K} c_{s_k} \tilde{x}_{v,s_k} + \sum_{l=2}^{L} c_t \tilde{x}_{t_l,v} - c_H \sum_{(i,j) \in A} \omega_{ij}^2 \tilde{x}_{ij} \right)
\]

\[
\leq \lambda g^H(\omega^1) + (1 - \lambda) g^H(\omega^2)
\]

The last inequality is because \( g^H(\omega) \) maximizes on \( \omega \) and \( \tilde{x} \) is feasible in both \( g^H(\omega^1) \) and \( g^H(\omega^2) \).

By the above lemmas, with similar deduction as in Theorem 3.2.3, we have

**Theorem 4.2.4** \( g^H(\omega) \leq h^H(\omega), \forall \omega \in [0, 1]^{|A|} \).

In summary, \( h^H(\omega) \) and \( g^H(\omega) \) have the following properties

P1 \( \forall \omega \in \{0, 1\}^{|A|}, h^H(\omega) = g^H(\omega). \)

P2 \( \forall \omega \in [0, 1]^{|A|}, h^H(\omega) \) is concave and \( g^H(\omega) \) is convex.

P3 \( \forall \omega \in [0, 1]^{|A|}, g^H(\omega) \leq h^H(\omega). \)

Now we can develop lower bounds as before, and integrate them in a branch and bound context. We will show that in details in Section 4.4.
4.3 Interdiction of Networks with Uncertain Sources and Destinations

In some interdiction problems, the attacker is not sure of the existences of sources and destinations, i.e. it only knows the probability of existences of sources/destinations. For example, the border police may not know whether there is any smuggling from $s_k$ and $t_k$. For problems like this, it's meaningful to extend the model to cover the networks with uncertain sources/destinations.

Consider a directed network $G(N, A)$ with $K$ sources and $L$ destinations, let $z$ be a $K + L$ dimension random binary vector with $z_i = 1$ indicates that the $i$th sources/destinations exists and $z_i = 0$ otherwise. Denote $Z$ as the space of $z$, which is finite and has the dimension of $2^{K+L}$. Let $p(z)$ be the probability of $z$ for any $z \in Z$, which is known to the attacker. The realization of $z$ is independent from the interdiction results.

Let $h(z, \omega)$ be the max-flow problem considering the existing sources/destinations in $z$ given the network state $\omega$, then

$$h(z, \omega) := \max_{x \in X^H(\omega)} \sum_{k=1}^{K} c_{s_k} z_{s_k, s_k} x_{s_k} + \sum_{l=1}^{L} c_{t_l} z_{t_l} x_{t_l, v}.$$ 

Compared with $h^H(\omega)$ in the multi-source/destination network

$$h^H(\omega) = \max_{x \in X^H(\omega)} \sum_{k=1}^{K} c_{s_k} x_{s_k, s_k} + \sum_{l=1}^{L} c_{t_l} x_{t_l, v},$$

$h(z, \omega)$ only considers the existing sources/destinations (i.e. sources/destinations with corresponding $z_i = 1$). Given a network state $\omega$, the network maximizes the expected maximum flow

$$h^Z(\omega) := \sum_{z \in Z} p(z) h(z, \omega) \quad (4.13)$$

Similarly, define $g(z, \omega)$ as the penalty problem considering the existing sources and
destinations in \( z \) given the network state \( \omega \)

\[
\max_{z \in X^H} \sum_{k=1}^{K} c_{sk} z_{k} x_{v,s_k} + \sum_{l=1}^{L} c_{tl} z_{l} x_{t_l,v} - c_{H} \sum_{(i,j) \in A} \omega_{ij} x_{ij}
\]

where \( c_{H} \geq \max_k c_{sk} + \max_l c_{tl} \). Compared with \( g^H(\omega) \) in the multi-source/destination network

\[
g^H(\omega) = \max_{z \in X^H(\omega)} \sum_{k=1}^{K} c_{sk} x_{v,s_k} + \sum_{l=1}^{L} c_{tl} x_{t_l,v} - c_{H} \sum_{(i,j) \in A} \omega_{ij} x_{ij},
\]

\( g(z, \omega) \) only considers the existing sources/destinations in \( z \). Define the penalty problem for the uncertain sources/destinations network as

\[
g^Z(\omega) := \sum_{z \in Z} p(z) g(z, \omega),
\]

Note that for any \( z \), one can take \( \tilde{c}_{sk} := c_{sk} z_{sk}, \tilde{c}_{tl} := c_{tl} z_{tl} \) as the new set of unit flow prices, then all the results that applied to \( h^H(\omega) \) and \( g^H(\omega) \) can be applied to \( h(z, \omega) \) and \( g(z, \omega) \). By theorem 4.2.1, for any \( z \),

\[
h(z, \omega) = g(z, \omega), \forall \omega \in \{0, 1\}^{|A|}.
\]

**Theorem 4.3.1** \( h^Z(\omega) = g^Z(\omega), \forall \omega \in \{0, 1\}^{|A|} \).

**Proof:** Since for any \( z \), \( h(z, \omega) = g(z, \omega) \), for any \( \omega \in \{0, 1\}^{|A|} \), then

\[
h^Z(\omega) = \sum_{z \in Z} p(z) h(z, \omega) = \sum_{z \in Z} p(z) g(z, \omega) = g^Z(\omega).
\]

**Lemma 4.3.2** For any \( \omega \in [0, 1]^{|A|} \), \( h^Z(\omega) \) is concave and \( g^Z(\omega) \) is convex.

**Proof:** For any \( z \), by lemma 4.2.2, \( h(z, \omega) \) is concave on \( \omega \in [0, 1]^{|A|} \), then \( h^Z(\omega) \) is also concave on \( \omega \in [0, 1]^{|A|} \) as the convex combination of \( h(z, \omega) \). Similarly, we can show \( g^Z(\omega) \) is convex on \( \omega \in [0, 1]^{|A|} \) due to the convexity of \( g(z, \omega) \).
Theorem 4.3.3 $g^Z(\omega) \leq h^Z(\omega), \forall \omega \in [0, 1]^{|A|}$.

Proof: Because the concavity of $h^Z(\omega)$ and the convexity of $g^Z(\omega)$, with similar deduction to 3.2.3, one can show

$$g^Z(\omega) \leq h^Z(\omega), \forall \omega \in [0, 1]^{|A|},$$

by using Jensen’s Inequality.

In summary, $h^Z(\omega)$ and $g^Z(\omega)$ have the following properties

P1 $\forall \omega \in \{0, 1\}^{|A|}, h^Z(\omega) = g^Z(\omega)$.

P2 $\forall \omega \in [0, 1]^{|A|}, h^Z(\omega)$ is concave and $g^Z(\omega)$ is convex.

P3 $\forall \omega \in [0, 1]^{|A|}, g^Z(\omega) \leq h^Z(\omega)$.

Now we can develop lower bounds as before, and integrate them in a branch and bound context. We will show the details in next section.

4.4 Solutions to the Extended Models

In this section we formulate the stochastic interdiction problems on a generalized network, which includes all cases discussed before. Then we provide solution method to this general model.

Let $h^*(\omega)$ be the optimization problem that the network maximizes given the network state $\omega$. It can be $h(\omega)$ in the basic model, $h^U(\omega)$ in the model with undirected network, $h^H(\omega)$ in the model with multi-source/destination network, and $h^Z(\omega)$ in the model with network with uncertain sources/destinations. Let $\gamma$ be the attacker’s strategy and $\Gamma(R)_b$ be the feasible set with $R$ as the resource constraint, then the interdiction problem is

$$J^* := \min_{\gamma \in \Gamma(R)_b} \sum_{I \in \Omega} P(I) h^*(I \cdot \gamma).$$ (4.14)
where $I$ is a possible scenario and $P(I)$ is the corresponding probability.

Let $g^*(\omega)$ be the corresponding penalty problem. It can be $g(\omega)$ in the basic model, $g^U(\omega)$ in the model with undirected network, $g^H(\omega)$ in the model with multi-source/destination network, and $g^Z(\omega)$ in the model with network with uncertain sources/destinations. As discussed before separately in these cases, $h^*(\omega)$ and $g^*(\omega)$ satisfy

P1 $\forall \omega \in \{0, 1\}^{|\Omega|}, h^*(\omega) = g^*(\omega)$.

P2 $\forall \omega \in [0, 1]^{|\Omega|}, h^*(\omega)$ is concave and $g^*(\omega)$ is convex.

P3 $\forall \omega \in [0, 1]^{|\Omega|}, g^*(\omega) \leq h^*(\omega)$.

Because of Property P1, $h^*(\omega)$ can be replaced by $g^*(\omega)$ in (4.14), then

$$J^* = \min_{\gamma \in \Gamma(R)_b} \sum_{I \in \Omega} P(I) g^*(I \cdot \gamma).$$

Because of Property P2, when we relax $\Gamma(R)_b$ to $\Gamma(R)$ and by Jensen’s Inequality, (4.14) has a lower bound of

$$\min_{\gamma \in \Gamma(R)} g^*(E[I] \cdot \gamma).$$

This may not be a good approximation to (4.14). However, one can improve this approximation by taking finer partitions on $\Omega$, the space of $I$, as the same approach we solve the basic model. The rest of this section provides foundation to extend the method for the basic model to solve the general model.

Given a partition $\Phi$ on $\Omega$, denote the lower bound problem as

$$L^*(\Phi, \Gamma(R)) := \min_{\gamma \in \Gamma(R)} \sum_{\phi \in \Phi} P(\phi) g^*(\rho(\phi) \cdot \gamma).$$

**Theorem 4.4.1** Let $\Phi^1, \Phi^2$ be partitions of $\Omega$ with $\Phi^1 \leq \Phi^2$, then for any non-empty $\Gamma$,

$$L^*(\Phi^1, \Gamma) \leq L^*(\Phi^2, \Gamma).$$
Proof: The proof is similar to that of Theorem 3.2.5. Notice the only difference is that $L(\cdot, \cdot)$ changes to $L^*(\cdot, \cdot)$ with $g(\cdot)$ replaced by $g^*(\cdot)$. Since $g^*(\cdot)$ is also convex as $g(\cdot)$, one can show the result for $L^*(\cdot, \cdot)$.

For any binary strategy $\gamma$, define its expected performance as

$$u^*(\gamma) := \sum_{I \in \Phi} P(I) h^*(I \cdot \gamma)$$

and its approximation value with partition $\Phi$ as

$$l^*(\Phi, \gamma) := \sum_{\phi \in \Phi} P(\phi) g^*(\rho(\phi) \cdot \gamma).$$

As defined in 3.2.6, let $\Phi(D)$ be a partition aligned with $D$ where $D$ is a subset of arcs.

**Theorem 4.4.2** If $\gamma$ is binary, then $l^*(\Phi(D(\gamma)), \gamma) = u^*(\gamma)$.

Proof: The proof is similar to that of Theorem 3.2.7, with the only difference is $h(\cdot)$ replaced by $h^*(\cdot)$, and $g(\cdot)$ replaced by $g^*(\cdot)$.

Based on Theorem 4.4.1 and Theorem 4.4.2, by replacing $L(\cdot, \cdot)$ with $L^*(\cdot, \cdot)$ and $g(\cdot)$ with $g^*(\cdot)$ in Algorithm 4, we extend our method for the general models, with details are shown in Algorithm 5.

### 4.5 Interdiction Problems on IEEE Bus 300 System

Now we apply MBB to solve the stochastic network interdiction problems on a power grid, the IEEE 300 Bus System, which is one of the power system test cases from (Christie, 1999).

The power system is modeled as an undirected flow network with multiple sources and destinations, where the generators, loads and buses are nodes and the connecting branches are arcs. The topology of IEEE 300 Bus System is shown in Fig. 4.1. The
**Procedure 5** Branch and Bound Algorithm for the Extended Models

1: Initialize $\gamma^* \leftarrow 0$, $J^* \leftarrow h^*(0)$, the optimal value without attack.
2: Solve $L^*([\emptyset], \Gamma(R))$, denote the optimal value $L^0$ and solution $\gamma^0$.
3: Initialize the set of candidate branch(es) $Q \leftarrow \{ (\gamma^0, L^0, \{\emptyset\}) \}$.
4: while $Q \neq \emptyset$ do
5:   Select and remove a node $(\gamma^k, L^k, \Gamma^k)$ from $Q$.
6:   if $\gamma^k$ is binary then
7:     Evaluate $u^k(\gamma^k) = \sum_{\phi \in \Phi(D(\gamma^k))} P(\phi) g^*(\rho(\phi) \cdot \gamma)$.
8:     if $u(\gamma^k) < J^*$ then
9:       Update $\gamma^* \leftarrow \gamma^k$, $J^* \leftarrow u(\gamma^k)$.
10:      Remove any node $(\tilde{\gamma}, \tilde{L}, \tilde{\Gamma})$ in $Q$ if $\tilde{L} \geq J^*$.
11:   end if
12: end if
13: Select an arc that have the maximum fraction interdiction $r_{ij}$,
\[ (i^*, j^*) \in \arg \max_{(i,j) \in A(k)} \gamma^k_{ij}. \]
14: Split $\Gamma^k$ on $(i^*, j^*)$ such that,
15: $\Gamma^{k,0} := \Gamma^k \cap \{ r_{i^*,j^*} = 0 \}$ and $\Gamma^{k,1} := \Gamma^k \cap \{ r_{i^*,j^*} = 1 \}$.
16: for $c = 0, 1$ do
17:   Solve $L^*(\Phi(D(\Gamma^{k,c})), \Gamma^{k,c})$, denote the optimal value $L^{k,c}$ and solution $\gamma^{k,c}$.
18:   if $L^{k,c} < J^*$ then
19:     add $(\gamma^{k,c}, L^{k,c}, \Gamma^{k,c})$ to $Q$.
20: end if
21: end for
22: end while
23: Output $\gamma^*, J^*$ as the optimal solution and value.
flow on a branch depends on the parameters of both ends and that branch. According to (Saccomanno, 2003), the flow on arc \((i,j)\) is

\[
f_{ij} = \frac{v_i v_j}{x_{ij}} \sin(\alpha_{ij}),
\]

where \(v_k, k = i, j\) is the voltage of node \(k\) and \(x_{ij}, \alpha_{ij}\) are the reactance and phase angle of arc \((i,j)\), respectively. We take \(\frac{1}{x_{ij}}\) as the capacity of arc \((i,j)\). Generators with degree of one are considered as sources, and other nodes with degree one are sinks. Branches with the same starting node and ending node are merged as one arc. The resulting network has 409 arcs and 300 nodes, out of which 33 are sources and 36 are sinks. We further adjust the capacities of arcs that directly linked to the sources and the destinations such that all demands in sinks are met given there is no attack on the network, which reflects the real supply and demand in the power grid. All the arcs within the network are attackable, with the same attacking costs (1 per arc) and the same probability of survival if attacked, 0.5. As before, we assume that the events of attacked arcs surviving are mutually independent.

For the interdiction problems on this multi-source, multi-sink, undirected network, we can first convert them to the problems of directed network and then further convert them to the problems of directed networks with multiple sources/destinations. Because the three properties of \(h^H(\omega), g^H(\omega)\) developed in Subsection 4.2, by replacing \(g(\cdot)\) with \(g^H(\cdot)\) in (2.9), and \(h(\cdot)\) with \(h^H(\cdot)\) in (2.10), Theorem 2.1.8 still holds. Therefore Sequential Approximation Algorithm (SAA, Algorithm 1) can be extended to solve the interdiction problems on these networks. Similarly replacing \(g(\cdot)\) with \(g^H(\cdot)\) in (2.16), Sample Averaging Approach (SAM) SAM can also be extended for these problems.

We compare Algorithm 5 with these two methods. Their running times (in terms of seconds) are shown in Tab. 4.1. The second row is the number of feasible attacking
strategies, which is a measure of the problem’s difficulty. The parameters for SAM are \((N, M, E) = (50, 10, 10^5)\) in SAM1 and \((N, M, E) = (5000, 10, 10^5)\) in SAM2, where \(N\) is the subproblems’ size, \(M\) is the subproblem’s number and \(E\) is the evaluation sample number. We ran SAM 50 times for each problem and show its average performance in Tab. 4.1.

MBB is substantially better than other methods in solving all these problems. Compared with SAA, MBB is about two orders of magnitude faster. In the case when \(R = 9\), MBB spends 107.11 seconds while SAA requires 14689.51 seconds. SAM takes less computation than SAA, but it’s still slower than MBB in about one
Table 4.1: Solving Interdiction Problems on IEEE BUS 300 Systems.

<table>
<thead>
<tr>
<th>Budget Str.#</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBB</td>
<td>0.67</td>
<td>2.26</td>
<td>2.68</td>
<td>23.96</td>
<td>107.11</td>
</tr>
<tr>
<td>SAA</td>
<td>1.91</td>
<td>5.23</td>
<td>6.30</td>
<td>521.37</td>
<td>14689.51</td>
</tr>
<tr>
<td>SAM1</td>
<td>213.53</td>
<td>221.96</td>
<td>244.53</td>
<td>308.29</td>
<td>480.96</td>
</tr>
<tr>
<td>err</td>
<td>0.09%</td>
<td>0.19%</td>
<td>0.21%</td>
<td>0.22%</td>
<td>0.32%</td>
</tr>
<tr>
<td>std</td>
<td>0.11%</td>
<td>0.22%</td>
<td>0.22%</td>
<td>0.24%</td>
<td>0.34%</td>
</tr>
<tr>
<td>SAM2</td>
<td>319.07</td>
<td>434.50</td>
<td>592.21</td>
<td>982.96</td>
<td>1411.17</td>
</tr>
<tr>
<td>err</td>
<td>0.02%</td>
<td>0.04%</td>
<td>0.07%</td>
<td>0.10%</td>
<td>0.12%</td>
</tr>
<tr>
<td>std</td>
<td>0.02%</td>
<td>0.05%</td>
<td>0.09%</td>
<td>0.11%</td>
<td>0.13%</td>
</tr>
</tbody>
</table>

order of magnitude. For the same case, SAM needs 480.96 seconds if its subproblems consider only 50 scenarios and 1411.17 if 5000 scenarios is considered.
Chapter 5

Multi-Stage Interdiction Problems

In this chapter, we study a new class of stochastic network interdiction problems where the attacker can attack multiple times and adapt its next attacks based on the observed outcomes of previous attacks. This is an extension of the previous stochastic network interdiction problem. The problem becomes more difficult due to the exponentially growing number of outcomes in each stage. We develop a model-predictive approach to tackle the difficulty. We focus on the two-stage interaction problem and develop a new set of performance bounds, which are integrated into a branch and bound procedure that extends the single stage approach to multiple stages. We also extend the model to undirected networks, multi-source/destination networks, uncertain source/destination networks and provide solution to these models.

5.1 Problem Formulation

Consider an interdiction problem on a directed network $G(N, A)$ where the attack takes place in two rounds. The first attack's outcomes are observed before the second attack is launched. After the second attack, the surviving network is used to conduct maximum flow from a source $s$ to a destination $t$. The problem consists of selecting the dynamic attacks in order to minimize the max-flow conducted by the network after the outcomes of both attacks take place. The sequence of the actions are shown in Fig. 5.1.

For $k = 1, 2$, for any $(i, j) \in A$, let $\gamma^k$ be the $k$th attack with $\gamma^k_{ij} = 1$ means
attacking arc \((i, j)\) and \(\gamma^k_{ij} = 0\) otherwise; let \(I^k\) be the outcome of \(\gamma^k\) with \(I^k_{ij} = 1\) means arc \((i, j)\) is removed given it’s attacked in \(\gamma^k\) and \(I^k_{ij} = 0\) otherwise. For simplicity, we assume that the outcomes of arcs being attacked are independent across arcs and stages. Moreover for each arc, the probability of successful attack is constant across the stages. We restrict each stage to include a maximum of one attack per arc, so \(\gamma^1, \gamma^2 \in \{0, 1\}^{|A|}\). Therefore for any arc \((i, j) \in A\), \(\omega^1_{ij} := \gamma^1_{ij}I^1_{ij}\) indicates its availability with \(\omega^1_{ij} = 1\) means arc \((i, j)\) is NOT available after the first attack and \(\omega^1_{ij} = 0\) otherwise. Similarly

\[
\omega^2_{ij} := 1 - (1 - \omega^1_{ij})(1 - \gamma^2_{ij}I^2_{ij})
\]

indicates the availability of arc \((i, j)\) after the second attack with \(\omega^2_{ij} = 1\) means arc \((i, j)\) is NOT available after both attacks and \(\omega^2_{ij} = 0\) otherwise. We allow \(\gamma^2\) to adapt to the observed first stage outcomes, i.e., \(\gamma^2\) depends on \(\gamma^1 \cdot I^1\), denoted as \(\gamma^2(\gamma^1 \cdot I^1)\). Because \(\gamma^1, I^1, \gamma^2(\gamma^1 \cdot I^1), I^2\) are binary-valued vectors, for any \((i, j)\), we have

\[
\omega^2_{ij} = \gamma^1_{ij}I^1_{ij} + \gamma^2_{ij}(\gamma^1 \cdot I^1)I^2_{ij} - \gamma^1_{ij}I^1_{ij}\gamma^2_{ij}(\gamma^1 \cdot I^1)I^2_{ij} = \gamma^1_{ij}I^1_{ij} + \gamma^2_{ij}(\gamma^1 \cdot I^1)I^2_{ij}
\]

(5.1)

where the last equality follows because, whenever \(\gamma^1_{ij}I^1_{ij} = 1\), arc \((i, j)\) is destroyed in stage 1, and thus gets no subsequent attack in stage 2, so \(\gamma^2_{ij}(\gamma^1 \cdot I^1) = 0\). Let the resource limits in the first stage and in the second stage be \(R^1, R^2\) respectively. Then

**Figure 5.1:** The sequence of actions of the two-stage interdiction problem.
the two-stage interdiction problem as

$$J_{2_{\text{org}}} := \min_{\gamma^1 \in \Gamma(R^1)} \sum_{I^1 \in \Omega} P(I^1) \min_{\gamma^2(\gamma^1 \cdot I^1) \in \Gamma(R^2)} \sum_{I^2 \in \Omega} P(I^2) h(\omega),$$

(5.2)

where $\omega$ are defined in (5.1). Notice $\omega$ is binary; By theorem 3.2.1, one can replace $h(\omega)$ with $g(\omega)$ in (5.2).

### 5.2 Model-Predictive Approach

Notice there are exponential number of realizations of $I^1$ and $I^2$, tremendous computations are required to evaluate a strategy over two stages. To make it tractable for a network with a reasonable size, we develop a model-predictive approach as in (Castañón and Wohletz, 2009).

Consider the following approximation problem

$$J_2 := \min_{\gamma^1 \in \Gamma(R^1)} \sum_{I^1 \in \Omega} P(I^1) \min_{\gamma^2(\gamma^1 \cdot I^1) \in \Gamma(R^2)} g(\gamma^1 \cdot I^1 + \gamma^2(\gamma^1 \cdot I^1) \cdot p)$$

(5.3)

where $p := E[I^2]$ is the vector of probabilities of successful interdiction on arcs. Compared with Problem (5.2), (5.3) replaces $\gamma^2(\gamma^1 \cdot I^1) \cdot I^2$ with its expectation $\gamma^2(\gamma^1 \cdot I^1) \cdot p$. Note also that (5.3) allows the second stage interdictions $\gamma^2(\gamma^1 \cdot I^1)$ to be continuous instead of binary-valued. A possible interpretation for this is to let $\gamma^2_{ij}(\gamma^1 \cdot I^1)$ be the probability to attack arc $(i, j)$ in the second stage if the network condition at that time is $\gamma^1 \cdot I^1$.

**Theorem 5.2.1** $J_2$ is a lower bound of $J_{2_{\text{org}}}$.

**Proof:** Due to the convexity of $g(\cdot)$, for any $\gamma^1, I^1$, the inner minimization
problem in (5.2) is

$$
\min_{\gamma^2 \in \Gamma(R^2)_b} \sum_{P^2 \in \Omega} P(I^2) g(\gamma^1 \cdot I^1 + \gamma^2 (\gamma^1 \cdot I^1) \cdot I^2)
$$

\[
\geq \min_{\gamma^2 \in \Gamma(R^2)_b} g(\gamma^1 \cdot I^1 + \gamma^2 (\gamma^1 \cdot I^1) \cdot (\sum_{P^2 \in \Omega} P(I^2)I^2))
\]

\[
= \min_{\gamma^2 \in \Gamma(R^2)_b} g(\gamma^1 \cdot I^1 + \gamma^2 (\gamma^1 \cdot I^1) \cdot p).
\]

Relaxing $\Gamma(R^2)_b$ to $\Gamma(R^2)$, we have

$$
\min_{\gamma^2(\gamma^1, I^1) \in \Gamma(R^2)} \sum_{P^2 \in \Omega} P(I^2) g(\gamma^1 \cdot I^1 + \gamma^2 (\gamma^1 \cdot I^1) \cdot I^2) \geq \min_{\gamma^2(\gamma^1, I^1) \in \Gamma(R^2)} g(\gamma^1 \cdot I^1 + \gamma^2 (\gamma^1 \cdot I^1) \cdot p).
$$

Notice the above equation is true for any $\gamma^1, I^1$, so summing over $I^1$ and minimizing $\gamma^1$ on both sides, we have $J2 \leq J2_{org}$. 

Compared with $J2_{org}$ where one has to consider all possible outcomes of both $I^1, I^2$, the lower bound problem $J2$ requires less computation since it only enumerates scenarios of $I^1$. In model-predictive approach, we first solve problem (5.3), whose optimal solution corresponding to the first attack $\gamma^1$ is binary. After carrying out the first attack $\gamma^1$ and observing the network surviving from it, we have a single-interdiction problem, which can be solved by our method in chapter 3.

For a model with more than two stage attacks, we can solve it in a similar approach, which considers fractional non-first-stage strategies and the average effect of the corresponding results. After the first attack and the network is updated, the resulting problem is one stage less. By doing this iteratively, one can have binary attacking strategies for all stages.

The success of the model-predictive approach depends on the quality of the lower bound approximation. Since we focus on the reduction of flows, comparing the expected flow reduction would be a good way to measure the quality of approximation. Given a first attack strategy $\gamma$, let $E(\gamma)$ be the expected max-flow after all attacks,
assuming the first attack is $\gamma$ and the sequential attacks are all binary and optimized based on the updated network. Define the expected flow reduction $r(\gamma)$ as

$$r(\gamma) := 1 - E(\gamma)/f,$$  \hspace{1cm} (5.4)

where $f$ is the max-flow in the network without attack.

In the following discussion, we show the approximation quality with numerical results of two-round interdiction problems on SNIP $7 \times 5$ (Fig. 3·3) and SNIP $4 \times 9$ (Fig. 3·4). In each problem, we evaluate the lower bound values for all first attack strategies $\gamma^1 \in \Gamma^1(R)$, and rank $\gamma^1$ by these values. We also calculate their flow reductions $r(\gamma^1)$. Specifically, to compute $E(\gamma^1)$ for each first attack strategy $\gamma^1$, we enumerate all possible outcome $I^1$ and for each updated network state $\gamma^1 \cdot I^1$, we solve a one-stage problem to get the minimum expected flow $f(\gamma^1 \cdot I^1)$. Then averaging over $I^1$ with its probability $P(I^1)$, we get $E(\gamma^1) := \sum_{I^1} P(I^1) f(\gamma^1 \cdot I^1)$.

Fig. 5·2 shows the flow reductions versus the approximation ranks for the top 1\% rank strategies. We solve four interdiction problems for each network (the top 4 plots for SNIP $7 \times 5$ and the bottom 4 plots for SNIP $4 \times 9$) with different attacking budgets in the first round and in the second round, denoted as $R^1, R^2$ respectively. Within each plot, the solid line is the flow reduction achieved by the optimal solution of the exact problem. Each point represents a strategy whose y-axis is the flow reduction and x-axis is the approximation rank, i.e., left strategies are better than right strategies in terms of the approximation values; upper strategies are better than lower strategies in terms of the exact values. A good approximation should have strategies' ranks closely match their reduction, which is the case for all problems shown in Fig. 5·2.
Figure 5.2: Flow reductions $r$ versus approximation rankings of the top 1% strategies in the approximation problems.
The optimal reductions are achievable by top ranked strategies. There are other interesting observations. First, the approximation becomes worse when the budgets in both stages increase, especially the second stage budget. For SNIP $7 \times 5$, the band of points in $(R^1, R^2) = (5, 3)$ becomes wider than that in $(R^1, R^2) = (5, 2)$, meaning the variance of flow reduction with similar approximation rank is larger. In SNIP $4 \times 9$, this effect is more obvious. Second, the bands in SNIP $7 \times 5$ are larger than those in SNIP $4 \times 9$, meaning the approximation is better in SNIP $7 \times 5$ than in SNIP $4 \times 9$. Notice SNIP $4 \times 9$'s min-cut contains only 4 arcs while SNIP $7 \times 5$'s min-cut has 7. The second observation suggests that the approximation is better when the network has larger min-cut.

We further compare the optimal approximation solution $\tilde{\gamma}$ with the exact optimal solution $\gamma^*$ in terms of flow reductions. The testing problems are on SNIP $7 \times 5$ and SNIP $4 \times 9$, with the first attack budget $R^1$ ranges from 5 to 9 and the second attack budget $R^2$ ranges 2 to 3. The results are shown in Tab. 5.1. In most cases, MBB finds the exact optimal solutions. Even in the worse case where the approximation solution is not the exact optimal solution, it still achieves more than 99.9% optimal reduction in SNIP $7 \times 5$ and more than 99.67% in SNIP $4 \times 9$.

### 5.3 Branch and Bound for the Approximation Problem

It's still a challenge to solve (5.3). First, the problem is combinatorial in size, where the number of the first attacks grows exponentially with the number of arcs. Second, the evaluation of any strategy requires a summation over an exponential number of outcomes, which would require solving an exponential number of min-max problems

$$
\min_{\gamma^2(\gamma^1 \cdot I^1) \in \Gamma(R^2)} g(\gamma^1 \cdot I^1 + \gamma^2(\gamma^1 \cdot I^1) \cdot p).
$$

We extend the MBB algorithm in chapter 3 to tackle these difficulties. Given $\Phi$
Table 5.1: The approximation quality of Model-Predictive Approach

<table>
<thead>
<tr>
<th>((R^1, R^2))</th>
<th>SNIP 7 × 5</th>
<th>SNIP 4 × 9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(r(\gamma^*))</td>
<td>(r(\bar{\gamma}))</td>
</tr>
<tr>
<td>(5,2)</td>
<td>78.39%</td>
<td>100%</td>
</tr>
<tr>
<td>(6,2)</td>
<td>84.90%</td>
<td>100%</td>
</tr>
<tr>
<td>(7,2)</td>
<td>89.44%</td>
<td>100%</td>
</tr>
<tr>
<td>(8,2)</td>
<td>91.74%</td>
<td>100%</td>
</tr>
<tr>
<td>(9,2)</td>
<td>92.73%</td>
<td>100%</td>
</tr>
<tr>
<td>(5,3)</td>
<td>84.87%</td>
<td>99.98%</td>
</tr>
<tr>
<td>(6,3)</td>
<td>89.65%</td>
<td>100%</td>
</tr>
<tr>
<td>(7,3)</td>
<td>92.64%</td>
<td>100%</td>
</tr>
<tr>
<td>(8,3)</td>
<td>94.11%</td>
<td>100%</td>
</tr>
<tr>
<td>(9,3)</td>
<td>94.78%</td>
<td>99.91%</td>
</tr>
</tbody>
</table>

as a partition of the space of \(I^1\), construct an approximation problem \(L2(\Phi, \Gamma(R^1))\) by averaging \(I^1\) and relaxing the binary constraints on \(\gamma^1\) in (5.3),

\[
L2(\Phi, \Gamma(R^1)) := \min_{\gamma^1 \in \Gamma(R^1)} \sum_{\phi \in \Phi} P(\phi) \min_{\gamma^2(\gamma^1, \rho(\phi)) \in \Gamma(R^2)} g(\gamma^1 \cdot \rho(\phi) + \gamma^2(\gamma^1 \cdot \rho(\phi)) \cdot p), \tag{5.5}
\]

where \(\gamma^2(\gamma^1 \cdot \rho(\phi))\) depends on \(\phi\). Because \(g(\cdot)\) is convex, by Jensen's Inequality, (5.5) is a lower bound of (5.3). Next we will show that the gap between the lower bound (5.5) and (5.3) can be tighten by refining partition \(\Phi\).

Given \(\gamma^1\) and \(\phi \in \Phi\), the inner minimization problem on \(\gamma^2\) in (5.5) is

\[
f(\gamma^1 \cdot \rho(\phi)) := \min_{\gamma^2 \in \Gamma R^2} g(\gamma^1 \cdot \rho(\phi) + \gamma^2 \cdot p),
\]

which is a min-max problem. Notice for any \(\omega \in [0, 1]^{\mathcal{A}}\), both of \(g(\omega)\) and its dual \(Dg(\omega)\) are bounded LP problems with feasible sets, there is strong duality between them. Therefore one can replace \(g(\omega)\) with \(Dg(\omega)\) in \(f(\gamma^1 \cdot \rho)\), which then becomes a
LP problem

\[
\begin{align*}
  &\min_{\gamma, \alpha \in \Gamma(R^2), \alpha, \pi} \sum_{(i,j) \in A} u_{ij} \alpha_{ij} \\
\text{s.t.} & \quad \pi - \pi_s \geq 1, \\
& \quad \alpha_{ij} + \pi_i - \pi_j + p\gamma_{ij}^2 \geq -\gamma_{ij}^1 \rho_{ij}, \\
& \quad \alpha_{ij} \geq 0, \forall (i,j) \in A
\end{align*}
\]

**Lemma 5.3.1** For any fixed $\gamma^1$, $f(\gamma^1 \cdot \rho)$ is convex on $\rho \in [0, 1]^{|A|}$.

**Proof:** For any $\rho^1, \rho^2 \in [0, 1]^{|A|}$, let their convex combination be

\[
\bar{\rho} := \lambda \rho^1 + (1 - \lambda) \rho^2, \forall \lambda \in [0, 1].
\]

Denote the corresponding optimal solutions to $f(\gamma^1 \cdot \rho)$ when $\rho = \rho^1, \rho^2$ as $\gamma^{2,1}, \gamma^{2,2}$ respectively. Let

\[
\bar{\gamma} := \lambda \gamma^{2,1} + (1 - \lambda) \gamma^{2,2}.
\]

Since in $f(\gamma^1 \cdot \bar{\rho})$, the feasible set is convex, then $\bar{\gamma}$ is feasible and therefore

\[
g(\gamma^1 \cdot \bar{\rho} + \bar{\gamma}^2 \cdot p) \geq \min_{\gamma^2 \in \Gamma(R^2)} g(\gamma^1 \cdot \bar{\rho} + \gamma^2 \cdot p) = f(\gamma^1 \cdot \bar{\rho}).
\]

Moreover, because the convexity of $g(\cdot)$,

\[
g(\gamma^1 \cdot \bar{\rho} + \bar{\gamma}^2 \cdot p) \\
= g(\lambda (\gamma^1 \cdot \rho^1 + \gamma^{2,1} \cdot p) + (1 - \lambda) (\gamma^1 \cdot \rho^2 + \gamma^{2,2} \cdot p)) \\
\leq \lambda g(\gamma^1 \cdot \rho^1 + \gamma^{2,1} \cdot p) + (1 - \lambda) g(\gamma^1 \cdot \rho^2 + \gamma^{2,2} \cdot p) \\
= \lambda \min_{\gamma^2 \in \Gamma(R^2)} g(\gamma^1 \cdot \rho^1 + \gamma^2 \cdot p) + (1 - \lambda) \min_{\gamma^2 \in \Gamma(R^2)} g(\gamma^1 \cdot \rho^2 + \gamma^2 \cdot p) \\
= \lambda f(\gamma^1 \cdot \rho^1) + (1 - \lambda) f(\gamma^1 \cdot \rho^2).
\]

Therefore

\[
f(\gamma^1 \cdot \bar{\rho}) \leq \lambda f(\gamma^1 \cdot \rho^1) + (1 - \lambda) f(\gamma^1 \cdot \rho^2), \forall \lambda \in [0, 1].
\]
i.e. \( f(\gamma \cdot \rho) \) is convex on \( \rho \).

The following theorem shows that refining the partitions \( \Phi \) helps to tighten the gap between the lower bound approximation and (5.3).

**Theorem 5.3.2** Let \( \Phi^1, \Phi^2 \) be partitions of \( l^1 \)'s space \( \Omega \). If \( \Phi^1 \leq \Phi^2 \), then for any non-empty \( \Gamma \),

\[
L2(\Phi^1, \Gamma) \leq L2(\Phi^2, \Gamma).
\]

**Proof:** Notice \( f(\gamma \cdot \rho) \) is convex on \( \rho \), then for any \( \gamma^1 \in \Gamma \),

\[
\sum_{\phi^2 \in \Phi^2} P(\phi^2) f(\gamma^1 \cdot \rho(\phi^2))
\]

\[
= \sum_{\phi^1 \in \Phi^1} P(\phi^1) \sum_{\phi^2 \in \Phi^1} \frac{P(\phi^2)}{P(\phi^1)} f(\gamma^1 \cdot \rho(\phi^2))
\]

\[
\geq \sum_{\phi^1 \in \Phi^1} P(\phi^1) f(\gamma^1) \cdot \sum_{\phi^2 \in \Phi^1} \frac{P(\phi^2)}{P(\phi^1)} \rho(\phi^2))
\]

\[
= \sum_{\phi^1 \in \Phi^1} P(\phi^1) f(\gamma^1 \cdot \rho(\phi^1)) .
\]

Since the above inequality is true for any \( \gamma^1 \), minimizing \( \gamma^1 \) over \( \Gamma \) on both sides, we have

\[
L2(\Phi^1, \Gamma) \leq L2(\Phi^2, \Gamma).
\]

For a binary strategy \( \gamma \), denote its lower bound value under partition \( \Phi \) as

\[
l^2(\Phi, \gamma) := \sum_{\phi \in \Phi} P(\phi) f(\gamma \cdot \phi)
\]

and its objective value in (5.3) as

\[
u^2(\gamma) := \sum_{I \in \Phi} P(I) f(\gamma \cdot I).
\]

Let \( \Phi(D(\gamma)) \) be the partition aligned with \( D \), as defined in 3.2.6, where \( D \) is a subset
of arcs. The following result shows that the gap between $u^2(\gamma)$ and $l^2(\Phi, \gamma)$ can be zero without enumerating all possible outcomes $I^1$, i.e. $\Phi \neq \Omega$.

**Theorem 5.3.3** If $\gamma$ is binary, then $l^2(\Phi(D(\gamma)), \gamma) = u^2(\gamma)$.

**Proof:** Since $\Phi(D(\gamma))$ is the full partition with respect to $\gamma$, then for any $\phi \in \Phi(D(\gamma))$, 

$$\gamma \cdot I = \gamma \cdot \phi, \ \forall I \in \phi,$$

which means $f(\gamma \cdot I) = f(\gamma \cdot \phi), \ \forall I \in \phi$. Also note $\sum_{I \in \phi} P(I) = P(\phi)$, then

$$u^2(\gamma) = \sum_{\phi \in \Phi(D(\gamma))} \sum_{I \in \phi} P(I) f(\gamma \cdot I) = \sum_{\phi \in \Phi(D(\gamma))} P(\phi) f(\gamma \cdot \phi) = l^2(\Phi(D(\gamma)), \gamma).$$

Based on theorem 5.3.2 and 5.3.3, our method can be extended to solve the approximation problem (5.3) by refining the partition $\Phi$ in $L2(\Phi, \Gamma)$. Details are shown in Algorithm 6.

### 5.4 Numerical Results

In this section, we implement our method as well as other two extended methods to solve two-stage network interdiction problems on SNIP $7 \times 5$ and SNIP $4 \times 9$ in a model-predictive approach, and then compare their performances in terms of running times as well as the quality of the output solutions. We also implement the brute force enumeration method (ENU) as a benchmark.

We evaluate the output strategies’ performances of different methods in a model-predictive way. Specifically, by implementing these methods for the two-stage problems, we have the binary strategies for the first interdiction. We carry out these strategies and for each possible outcome, it becomes a one-stage problem, which can be solved exactly by the methods developed in the previous chapters. We evaluate
Procedure 6 Modified B&B Method for Two-Stage Interdiction

1: Initialize $\gamma^* \leftarrow 0$, $J^* \leftarrow \text{max-flow without attack.}$
2: Solve $L2(\{\Omega\}, \Gamma(R))$, denote the optimal value $L^0$ and solution $\gamma^0$.
3: Initialize the set of candidate branch(es) $Q \leftarrow \{(\gamma^0, L^0, \{\Omega\})\}$.
4: while $Q \neq \emptyset$ do
5: Select and remove a node $(\gamma^k, L^k, \Gamma^k)$ from $Q$.
6: if $\gamma^k$ is binary then
7: Evaluate $u^2(\gamma^k) = \sum_{\phi \in \Phi(D(\gamma^k))} P(\phi) f(\gamma \cdot I)$.
8: if $u(\gamma^k) < J^*$ then
9: Update $\gamma^* \leftarrow \gamma^k$, $J^* \leftarrow u^2(\gamma^k)$.
10: Remove any node $(\gamma, \tilde{L}, \tilde{\Gamma})$ in $Q$ if $\tilde{L} \geq J^*$.
11: end if
12: end if
13: Select an arc that have the maximum fraction interdiction $r_{ij}^k$:

$$(i^*, j^*) \in \arg \max_{(i,j) \notin A(\Gamma^k)} \gamma_{ij}^k.$$

14: Split $\Gamma^k$ on $(i^*, j^*)$ such that
15: $\Gamma^{k,0} := \Gamma^k \cap \{r_{i^*,j^*} = 0\}$ and $\Gamma^{k,1} := \Gamma^k \cap \{r_{i^*,j^*} = 1\}$.
16: for $c = 0, 1$ do
17: Solve $L2(\Phi(D(\Gamma^{k,c})), \Gamma^{k,c})$, denote the optimal value $L^{k,c}$ and solution $\gamma^{k,c}$.
18: if $L^{k,c} < J^*$ then
19: add $(\gamma^{k,c}, L^{k,c}, \Gamma^{k,c})$ to $Q$.
20: end if
21: end for
22: end while
23: Output $\gamma^*, J^*$ as the optimal solution and value.
the expected performances of these strategies in terms of the reduced flow \( r(\gamma) \) as defined in (5.4).

**SAA Extension for two-stage problems** Cormican et al. (Cormican et al., 1998) extended their model to have multi-attacks on an arc. However, the attacker can not observe the first attack result to optimize its second attack. In their extended model, The problem can be formulated as

\[
\min_{\gamma^1, \gamma^2 \in \Gamma(R^1, R^2)} \sum_{I^1 \in \Omega, I^2} P(I^1) P(I^2) h(\omega(I^1, I^2))
\]

where \( \omega(I^1, I^2) \) is the network state after both attacks, which is

\[
\omega(I^1, I^2) := I^1 * \gamma^1 + (1 - I^1 \gamma^1) * I^2 \gamma^2.
\]

The constraints on \( \gamma^1, \gamma^2 \), represented by the feasible set \( \Gamma(R^1, R^2)_b \), are

\[
\gamma^1, \gamma^2 \in \{0, 1\}^{|A|}; \quad \sum_{(i,j) \in A} c_{ij}^1 \gamma^1_{ij} \leq R^1;
\]

\[
\sum_{(i,j) \in A} c_{ij}^2 \gamma^2_{ij} \leq R^2; \quad \gamma^1_{ij} \geq \gamma^2_{ij}, \forall (i, j) \in A
\]

where the last sets of inequality is to forbid taking one attack as the second one instead of the first one. Notice in this formulation, the second attack does NOT depends on the outcomes of the first attack \( I^1 \), which is the fundamental difference from our model. The extended the Sequence Approximation Algorithm (SAA) for this model is quite similar to the basic one, which partitions the space of the outcome \( I^1, I^2 \) to get sequentially improved lower bounds. Within a cell, let \( \rho^1, \rho^2 \) be the average outcomes of \( I^1, I^2 \) of this cell, then the averaged (fractional) network state
given strategies $\gamma^1, \gamma^2$ is

$$
\omega^f = \rho^1 \gamma^1 + (1 - \rho^1 \gamma^1) \rho^2 \gamma^2 = \rho^1 \gamma^1 + (1 - \rho^1) \rho^2 \gamma^2.
$$

(5.7)

The last equation is true due to the constraint of $\gamma^1 \geq \gamma^2$. Substitute $\omega^f$ into (2.9) and (2.10) to replace $\rho \cdot \gamma$, we have

$$
LBMIN^2(\Phi) := \min_{\gamma^1, \gamma^2 \in \Gamma(R^1, R^2)} \sum_{\phi^1 \in \Phi^1, \phi^2 \in \Phi^2} g(\rho(\phi^1) \cdot \gamma^1 + (1 - \rho(\phi^1)) \rho(\phi^2) \gamma^2),
$$

(5.8)

$$
UB^2(\Phi, \gamma) := \sum_{\phi^1 \in \Phi^1, \phi^2 \in \Phi^2} h(\rho(\phi^1) \cdot \gamma^1 + (1 - \rho(\phi^1)) \rho(\phi^2) \gamma^2).
$$

(5.9)

The details of SAA extension can be found in Algorithm 7. The partitioning subroutine is quite similar to the single-attack problem, except it restricts the partition of $I^2_{ij}$ must after the partition of $I^1_{ij}$ for any arc $(i, j)$ in $A$.

We compare MBB with this extended SAA method in Tab. 5.2 with the same interdiction problems as in previous examples. SAA is slower than MBB in about 1 to 2 orders of magnitude. In the case of SNIP $7 \times 5$ with budget 8, MBB runs about 5 seconds, compared with 216 seconds in SAA. More important, the solution obtained

---

**Procedure 7** Sequential Approximation Algorithm (SAA) for multiple attacks

1: Let $\Phi = \{\Omega\}$, $U^* = \inf$ and $L^* = 0$.
2: Solve $LBMIN^2(\Phi)$, denote the optimal value as $L^*$ and the solution as $\hat{\gamma}$.
3: if $U^* - L^* \leq \epsilon$ then
4: Output solution $\gamma^*$ and terminate.
5: end if
6: Evaluate $U' := UB^2(\Phi, \hat{\gamma})$.
7: if $U' < U^*$ then
8: Update $\gamma^* \leftarrow \hat{\gamma}$ and $U^* \leftarrow U'$.
9: end if
10: if $U^* - L^* \leq \epsilon$ then
11: Output solution $\gamma^*$ and terminate.
12: end if
13: Refine partition $\Phi$ according to partition procedure described below and go to Step 2.
by MBB is much better than those output by SAA. In the case of SNIP 7 × 5 with budget 5, MBB finds the exact solution but SAA outputs solution with performance only 83.58% of MBB’s solution.

**SAM Extension for two-stage problems** In a model predictive approach, we need to solve the approximation problem of (5.3) for the two-stage interdiction problem

\[
\min_{\gamma^1 \in \Gamma(R^1)} \sum_{I^1 \in \Omega} P(I^1) \min_{\gamma^2(\gamma^1 \cdot I^1) \in \Gamma(R^2)} g(\gamma^1 \cdot I^1 + \gamma^2(\gamma^1 \cdot I^1) \cdot p)
\]

As discussed before, the inner minimization problems can be replaced by \( f(\gamma^1 \cdot I^1) \), therefore we solve

\[
\min_{\gamma^1 \in \Gamma(R^1)} \sum_{I^1 \in \Omega} P(I^1) f(\gamma^1 \cdot I^1). \tag{5.10}
\]
Notice this is a stochastic programming problem, we can extend the Sample Averaging Approach (SAM) approach to solve it. Details are shown in Procedure 8. As shown before in Section 2.3, \( \bar{L} := \frac{1}{M} \sum_{m} L^m \) is the lower bound and \( \bar{U} := \frac{1}{M} \sum_{m} U^m \) is the upper bound in statistical sense. As in the single-attack problems, we implement SAM with different subproblems' sizes. In SAM1, the subproblem considers 50 samples. In SAM2, the subproblem considers 5000 samples. In both cases, we solve 10 subproblems and evaluate their solutions with 10000 samples.

We compare the performance of SAM1, SAM2 with our MBB algorithm of solving the two-stage interdiction problems on networks SNIP 7 × 5 and SNIP 4 × 9 in Tab. 5.3. In these problems, the first round budget ranges from 5 to 9 and the second round budget is fixed on 2. MBB is about 2 to 3 orders of magnitude faster than SAM1. Compared to SAM2, MBB is even faster. In the case of SNIP 7 × 5 with budget 5, MBB spends 0.27 second while SAM1 requires about 50 seconds. The running time of SAM2 are even more, about 115.44 seconds. With less time, MBB outputs solutions with higher quality, as measure by the reduced flow. In all cases of

### Procedure 8 Sample Averaging Approach (SAM) for Problem (5.10)

1. for \( m = 1, \ldots, M \) do
2. Randomly sample \( N \) scenarios based on the distribution of \( I \), denote the set of samples as \( S^m := \{ I^m_n \} \).
3. Solve \( \min_{\gamma^1 \in \Gamma^1} \frac{1}{M} \sum_{I^m \in S^m} f(\gamma^1 \cdot I^1) \). Denote the optimal solution and value as \( \gamma^m \) and \( L^m \) respectively.
4. end for
5. Randomly sample \( E \) scenarios based on the distribution of \( I \), denote the set of samples as \( S^E := \{ I^E_n \} \).
6. Initialize \( \gamma^* \leftarrow 0, U^* \leftarrow \infty \).
7. for \( m = 1, \ldots, M \) do
8. Calculate \( U^m := \frac{1}{E} \sum_{I^E \in S^E} f(\gamma^m \cdot I^E) \).
9. if \( U^* > U^m \) then
10. update \( \gamma^* \leftarrow \gamma^m, U^* \leftarrow U^m \).
11. end if
12. end for
13. Output \( \gamma^*, v^* \) as the optimal solution.
Table 5.3: Compare with SAM in model-predictive approach, where $\gamma^1, \gamma^2$ are solutions output by SAM1, SAM2.

<table>
<thead>
<tr>
<th>Budget</th>
<th>ENU2</th>
<th>MBB2</th>
<th>SAM1</th>
<th>SAM2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>$r(\gamma)$</td>
<td>$r(\gamma^*)$</td>
<td>$r(\gamma)$</td>
<td>$r(\gamma^*)$</td>
</tr>
<tr>
<td>5</td>
<td>53.68</td>
<td>0.27</td>
<td>100%</td>
<td>49.22</td>
</tr>
<tr>
<td>6</td>
<td>315.51</td>
<td>0.81</td>
<td>100%</td>
<td>52.39</td>
</tr>
<tr>
<td>7</td>
<td>1598.57</td>
<td>1.73</td>
<td>100%</td>
<td>60.22</td>
</tr>
<tr>
<td>8</td>
<td>6492.84</td>
<td>4.59</td>
<td>100%</td>
<td>90.78</td>
</tr>
<tr>
<td>9</td>
<td>22885.70</td>
<td>13.71</td>
<td>100%</td>
<td>241.21</td>
</tr>
<tr>
<td></td>
<td>SNIP 7 x 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>53.68</td>
<td>1.08</td>
<td>99.61%</td>
<td>65.49</td>
</tr>
<tr>
<td>6</td>
<td>315.51</td>
<td>3.37</td>
<td>100%</td>
<td>104.66</td>
</tr>
<tr>
<td>7</td>
<td>1598.57</td>
<td>15.99</td>
<td>99.99%</td>
<td>163.59</td>
</tr>
<tr>
<td>8</td>
<td>6492.84</td>
<td>57.69</td>
<td>99.67%</td>
<td>321.83</td>
</tr>
<tr>
<td>9</td>
<td>22885.70</td>
<td>180.02</td>
<td>100%</td>
<td>581.39</td>
</tr>
<tr>
<td></td>
<td>SNIP 4 x 9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SNIP $7 \times 5$, MBB find the exact optimal solutions. while the output of SAM1 can be as worse as 84.52% in the case of budget 7. In SNIP $4 \times 9$, it’s similar similar situation, the outputs of MBB are much closer to the optimal solution in terms of performance than the solutions obtained by SAM.

### 5.5 Extended Multi-stage Interdiction Problems

As discussed before, it’s meaningful to extend the basic model to cover different types of networks. In this section, we extend the model to emphasize multi-stage interdiction problems on undirected networks, multi-source/destination networks as well as the networks with uncertain source/destination. Notice in the basic multi-stage interdiction model, our method depends on the convexity of the inner minimization problem $f(\gamma \cdot \rho)$. We will formulate the corresponding problem in each extended model and show its convexity. Then we provide solutions to these models in a gen-
eral form. Finally we implement our method to solve some two-stage interdiction problems on IEEE BUS 300 System.

5.5.1 Interdiction of Undirected Networks

Given an undirected network $G(N, A^U)$, one can convert it into an equivalent directed network. Follow the discussion in Section 4.1 and adopt the formulation of $g^U(\omega)$,

$$g^U(\omega) := \max_{x \in X^U} x_{ts} - \sum_{(i,j) \in A^U} \omega_{ij} (x_{ij} + x_{ji}),$$

(5.11)

where $\omega$ is the network state after all attacks. Then the two-stage interdiction problem on the undirected network is

$$\min_{\gamma^1 \in \Gamma(R^1), \gamma^2 \gamma^2(\gamma^1 \cdot I^1), I^2} \sum_{I^1 \in \Omega} P(I^1) \min_{\gamma^2 \gamma^2(\gamma^1 \cdot I^1), I^2} \sum_{I^2 \in \Omega} P(I^2) g^U(\gamma^1 \cdot I^1 + \gamma^2(\gamma^1 \cdot I^1) \cdot I^2).$$

where $\gamma^1, I^1, \gamma^2(\gamma^1 \cdot I^1), I^2$ are defined as the same in the basic model in Section 5.1. As in the basic multi-stage model, in a model-predictive approach we will solve the following lower bound approximation,

$$\min_{\gamma^1 \in \Gamma(R^1), \gamma^2 \gamma^2(\gamma^1 \cdot I^1), I^2} \sum_{I^1 \in \Omega} P(I^1) \min_{\gamma^2 \gamma^2(\gamma^1 \cdot I^1), I^2} g^U(\gamma^1 \cdot I^1 + \gamma^2(\gamma^1 \cdot I^1) \cdot I^2),$$

which relaxes $\gamma^2$ to be fractional and takes the average effect of the second attack $\gamma^2 \cdot I^2$. Define the inner minimization problem as

$$f^U(\gamma^1 \cdot I) := \min_{\gamma^2 \in \Gamma(R^2)} g^U(\gamma^1 \cdot I + \gamma^2 \cdot I^2).$$
Notice the dual of $g^U(\omega)$ can be written as

$$
\begin{align*}
Dg^U(\omega) := \min_{\pi, \alpha} & \sum_{(i,j) \in A^U} u_{ij}(\alpha_{ij} + \alpha_{ji}) \\
\text{s.t.} & \quad \pi_t - \pi_s \geq 1, \\
& \quad \alpha_{ij} + \pi_i - \pi_j \geq -\omega_{ij}, \\
& \quad \alpha_{ji} + \pi_j - \pi_i \geq -\omega_{ij}, \\
& \quad \alpha_{ij}, \alpha_{ji} \geq 0, \forall (i,j) \in A^U
\end{align*}
$$

where $\pi_n, \forall n \in N$ is the dual variable corresponds to the flow conservation constraint (4.1) on node $n$, and $\alpha_{ij}, \alpha_{ji}, \forall (i,j) \in A^U$ corresponds to the capacity constraints (4.5). Since for any $\omega \in [0,1]^{|A^U|}$, $g^U(\omega)$ is feasible and bounded, then strong duality holds. By replacing $g^U(\cdot)$ with $Dg^U(\cdot)$, $f^U(\gamma^1 \cdot \rho)$ can be written as

$$
\begin{align*}
\min_{\gamma \in \Gamma(\mathbb{R}^2), \alpha, \pi} & \sum_{(i,j) \in A^U} u_{ij}(\alpha_{ij} + \alpha_{ji}) \\
\text{s.t.} & \quad \pi_t - \pi_s \geq 1, \\
& \quad \alpha_{ij} + \pi_i - \pi_j + p_{ij} \gamma_{ij}^2 \geq -\gamma_{ij}^1 \rho_{ij}, \\
& \quad \alpha_{ji} + \pi_j - \pi_i + p_{ij} \gamma_{ji}^2 \geq -\gamma_{ij}^1 \rho_{ij}, \\
& \quad \alpha_{ij}, \alpha_{ji} \geq 0, \forall (i,j) \in A^U
\end{align*}
$$

Lemma 5.5.1 For any fixed $\gamma^1 \in \{0,1\}^{|A^U|}$, $f^U(\gamma^1 \cdot \rho)$ is convex on $\rho$.

Proof: For any $\rho^1, \rho^2 \in [0,1]^{|A^U|}$, let their convex combination be

$$
\bar{\rho} := \lambda \rho^1 + (1 - \lambda) \rho^2, \forall \lambda \in [0,1].
$$

Denote the corresponding optimal solutions to $f^U(\gamma^1 \cdot \rho)$ when $\rho = \rho^1, \rho^2$ as $\gamma^{2,1}, \gamma^{2,2}$ respectively. Let $\gamma^{2} := \lambda \gamma^{2,1} + (1 - \lambda) \gamma^{2,2}$. Note in $f^U(\gamma^1 \cdot \bar{\rho})$, the feasible set is
convex, then $\gamma_2^2$ is feasible and

$$g^U(\gamma^1 \cdot \bar{p} + \gamma_2^2 \cdot p) \geq \min_{\gamma^2 \in \Gamma(R^2)} g^U(\gamma^1 \cdot \bar{p} + \gamma^2 \cdot p) = f^U(\gamma^1 \cdot \bar{p}).$$

Moreover, because the convexity of $g^U(\cdot)$,

$$g^U(\gamma^1 \cdot \bar{p} + \gamma_2^2 \cdot p)$$

$$\leq \lambda g^U(\gamma^1 \cdot \rho^1 + \gamma_2^2 \cdot p) + (1 - \lambda) g^U(\gamma^1 \cdot \rho^2 + \gamma_2^2 \cdot p)$$

$$= \lambda \min_{\gamma^2 \in \Gamma(R^2)} g^U(\gamma^1 \cdot \rho^1 + \gamma^2 \cdot p) + (1 - \lambda) \min_{\gamma^2 \in \Gamma(R^2)} g^U(\gamma^1 \cdot \rho^2 + \gamma^2 \cdot p)$$

$$= \lambda f^U(\gamma^1 \cdot \rho^1) + (1 - \lambda) f^U(\gamma^1 \cdot \rho^2).$$

Therefore $f^U(\gamma^1 \cdot \bar{p}) \leq \lambda f^U(\gamma^1 \cdot \rho^1) + (1 - \lambda) f^U(\gamma^1 \cdot \rho^2), \forall \lambda \in [0, 1].$

5.5.2 Interdiction of Multi-source/destination Networks

Consider a directed network $G(N, A)$ with nodes $N$ and arcs $A$ that transports homogeneous goods from $K$ sources to $L$ destinations, which are denoted as $s_k, t_l \in N$ with $k = 1, \ldots, K; l = 1, \ldots, L$, respectively. Let $c_{s_k}$ (or $c_{t_l}$) denotes the unit price for flows from $s_k$ (or to $t_l$). Follow the discussion in Section 4.2 and adopt the formulation of $g^H(\omega)$,

$$g^H(\omega) := \max_{x \in X^H} \sum_{k=1}^{K} c_{s_k} x_{s_k,v} + \sum_{l=1}^{L} c_{t_l} x_{t_l,v} - c_H \sum_{(i,j) \in A} \omega_{ij} x_{ij}. \quad (5.12)$$

where $\omega$ is the network state after all attacks. Then the two-stage interdiction problem is

$$\min_{\gamma^1 \in \Gamma(R^1)_b} \sum_{I_1} P(I_1) \min_{\gamma_2 \in \Gamma(R^2)_b} \sum_{I_2} P(I_2) g^H(\gamma^1 \cdot I_1^1 + \gamma^2 \cdot p).$$
In the model-predictive approach, we want to solve the following lower bound approximation

\[
\min_{\gamma^1 \in \Gamma(\mathbb{R}^1)} \sum_{I^1 \in \Omega} P(I^1) \quad \min_{\gamma^2 \in \Gamma(\mathbb{R}^2)} g^H(\gamma^1 \cdot I^1 + \gamma^2 \cdot p).
\]

Define the inner minimization problem as

\[
f^H(\gamma^1 \cdot I) := \min_{\gamma^2 \in \Gamma(\mathbb{R}^2)} g^H(\gamma^1 \cdot I + \gamma^2 \cdot p).
\]

The dual of \(g^H(\omega)\) can be written as \(Dg^H(\omega)\)

\[
\begin{align*}
\min_{\pi, \alpha} & \quad \sum_{(i,j) \in A} u_{ij} \alpha_{ij}, \\
\text{s.t.} & \quad \pi_v - \pi_{s_k} \geq c_{s_k}, \quad k = 1, \ldots, K, \\
& \quad \pi_{t_l} - \pi_v \geq c_{t_l}, \quad l = 1, \ldots, L \\
& \quad \alpha_{ij} + \pi_i - \pi_j \geq c_{H \omega_{ij}}, \forall (i, j) \in A \\
& \quad \alpha_{ij} \geq 0, \forall (i, j) \in \overline{A}
\end{align*}
\]

where \(\pi_n, \forall n \in \overline{N}\) is the dual variable corresponds to the flow conservation constraint (4.9) on node \(n\), and \(\alpha_{ij}, \forall (i, j) \in \overline{A}\) corresponds to the capacity constraint (4.11).

Since for any \(\omega \in [0, 1]^{|A|}\), \(g^H(\omega)\) is feasible and bounded, then by strong duality, replacing \(g^H(\omega)\) with \(Dg^H(\omega)\) in \(f^H(\gamma^1 \cdot \rho)\), we have

\[
\begin{align*}
\min_{\gamma^2 \in \Gamma(\mathbb{R}^2), \alpha, \pi} & \quad \sum_{(i,j) \in A} u_{ij} \alpha_{ij}, \\
\text{s.t.} & \quad \pi_v - \pi_{s_k} \geq c_{s_k}, \quad k = 1, \ldots, K, \\
& \quad \pi_{t_l} - \pi_v \geq c_{t_l}, \quad l = 1, \ldots, L \\
& \quad \alpha_{ij} + \pi_i - \pi_j + c_{H \rho_{ij} \gamma^2_{ij}} \geq -c_{\gamma^1_{ij} \rho_{ij}}, \forall (i, j) \in A \\
& \quad \alpha_{ij} \geq 0, \forall (i, j) \in \overline{A}
\end{align*}
\]

**Lemma 5.5.2** For any fixed \(\gamma^1\), \(f^H(\gamma^1 \cdot \rho)\) is convex on \(\rho \in [0, 1]^{|A|}\).
Proof: Similar to that of lemma 5.5.1 by utilizing the convexity of $g^H(\cdot)$. \hfill \blacksquare

5.5.3 Interdiction of Networks with Uncertainties Source/destination

Consider a directed network $G(N, A)$ with $K$ sources and $L$ destinations, let $z$ be a $K + L$ dimension random binary vector with $z_i = 1$ indicates that the $i$th source/destination exists and $z_i = 0$ otherwise. Follow the discussion in Section 4.3, and adopt the formulation of $g^Z(\omega) := \sum_{z \in \mathbb{Z}} p(z) g(z, \omega)$ with $g(z, \omega)$ as the penalty problem considering the existing source/destination in $z$ with the network state $\omega$

$$g(z, \omega) := \max_{x \in X_H} \sum_{k=1}^{K} c_{sk} z_{sk} x_{v, sk} + \sum_{l=1}^{L} c_{li} z_{li} x_{l, vi} - c_H \sum_{(i,j) \in A} \omega_{ij} x_{ij}.$$ 

Then the two-stage interdiction problem is

$$\min_{\gamma \in \Gamma(R^1)} \sum_{I \in \Omega} \sum_{P \in \Omega} P(I) \min_{\gamma^2(\gamma^1 \cdot I^1) \in \Gamma(R^2)} \sum_{I^2 \in \Omega} \sum_{P \in \Omega} P(I^2) g^Z(\gamma^1 \cdot I^1 + \gamma^2(\gamma^1 \cdot I^1) \cdot p).$$

The corresponding lower bound problem in the model-predictive approach is

$$\min_{\gamma \in \Gamma(R^1)} \sum_{I \in \Omega} \sum_{I^1 \in \Omega} P(I) g^Z(\gamma \cdot I + \gamma^2(\gamma \cdot I^1) \cdot p).$$

Define the inner minimization problem $f^Z(\gamma^1, \rho)$ as

$$\min_{\gamma^2 \in \Gamma(R^2)} g^Z(\gamma^1 \cdot I + \gamma^2 \cdot p).$$

Lemma 5.5.3 For any fixed $\gamma^1$, $f^Z(\gamma^1 \cdot \rho)$ is convex on $\rho \in [0, 1]^{|A|}$.

Proof: Similar to that of lemma 5.5.1 by utilizing the convexity of $g^Z(\cdot)$. \hfill \blacksquare

5.5.4 Solution to the Extended Models

We will extend our method for the basic model to solve these extended models. Denote $f^*(\gamma \cdot I)$ be the network’s optimization problem given the attacker’s first action $\gamma$ and
the corresponding outcome $I$. $f^*(\gamma \cdot I)$ can be $f(\gamma \cdot I)$ in the basic model; $f^U(\gamma \cdot I)$ for undirected networks; $f^H(\gamma \cdot I)$ for multi-source/destination networks, and $f^Z(\gamma \cdot I)$ for uncertain sources/destinations networks. Then the two-stage lower bound problem is

$$J^* := \min_{\gamma \in \Gamma(R)} \sum_{I \in \Omega} P(I) f^*(\gamma \cdot I).$$

(5.13)

Define the lower bound approximation with the partition $\Phi$ in $I$’s space as

$$L^2(\Phi, \Gamma(R)) := \min_{\gamma^1 \in \Gamma(R)} \sum_{\phi \in \Phi} P(\phi) f^*(\gamma \cdot \phi).$$

(5.14)

**Theorem 5.5.4** Let $\Phi^1, \Phi^2$ be partitions of $I$’s space $\Omega$. If $\Phi^1 \leq \Phi^2$, then for any non-empty set $\Gamma$,

$$L^2(\Phi^1, \Gamma) \leq L^2(\Phi^2, \Gamma).$$

Proof: The proof is the same as the basic model, which utilizes the convexity of $f^*(\gamma \cdot \rho)$ on $\rho \in [0,1]^{[\Gamma]}$. $\blacksquare$

For a binary strategy $\gamma$, denote its lower bound value with respect to partition $\Phi$ as

$$l^2(\Phi, \gamma) := \sum_{\phi \in \Phi} P(\phi) f^*(\gamma \cdot \phi)$$

and its expected performance as

$$u^2(\gamma) := \sum_{I \in \Phi} P(I) f^*(\gamma \cdot I).$$

We can show that the lower bound approximation can be tight without enumerating all possible outcomes $I$. Let $\Phi(D(\gamma))$ be the partition aligned with $D$, as defined in 3.2.6, where $D$ is a subset of arcs.

**Theorem 5.5.5** For any binary $\gamma$, $l^2(\Phi(D(\gamma)), \gamma) = u^2(\gamma)$.

Proof: For any $\phi \in \Phi(D(\gamma))$ and any $I \in \phi$, if $\gamma_{ij} = 0$, then

$$\rho(\phi)_{ij} \gamma_{ij} = I_{ij} \gamma_{ij} = 0.$$
If $\gamma_{ij} = 1$, since every scenario $I$ in $\phi$ has the same outcome in arc $(i, j)$, then

$$\rho(\phi)_{ij} \gamma_{ij} = I_{ij} \gamma_{ij}.$$ 

As a result,

$$\rho(\phi) \cdot \gamma = I \cdot \gamma, \forall I \in \phi.$$ 

Note also that $P(\phi) = \sum_{I \in \phi} P(I)$. These two observations imply that

$$P(\phi) f^*(\rho(\phi) \cdot \gamma) = \sum_{I \in \phi} P(I) f^*(I \cdot \gamma) = \sum_{I \in \phi} P(I) f^*(I \cdot \gamma).$$

Summing over all $\phi \in \Phi(D(\gamma))$, we have $l_2^* (\Phi(D(\gamma)), \gamma) = u_2^* (\gamma)$.

To solve the extended models, one just need to replace $f^*(\gamma \cdot \phi)$ with $f(\gamma \cdot \phi)$ in Algorithm 6. Here we implement MBB to solve the two-stage interdiction problems on IEEE 300 Bus System, which is formulated as an undirected network with multiple sources and destinations. In the second round the attacker at most interdicts two arcs. ENU cannot handle such problems because of the large size of the network. Therefore we only compare MBB with SAM in Tab. 5.4. The second row is the number of all possible (first stage) attacking strategies, which is a measure of the problem’s difficulty. The parameters for SAM are $(N, M, E) = (50, 10, 10^5)$ in SAM1 and $(N, M, E) = (5000, 10, 10^5)$ in SAM2. MBB is much faster than SAM, in about two orders of magnitude. In the case where $R^1 = 5$, MBB runs about 4 seconds while the fast SAM1 needs more than 800 seconds, and the slow SAM2 requires 1438 seconds. For problems with larger $R^1$, MBB is still much faster than SAM. When $R^1 = 9$, the running time of MBB is 32.52 seconds, comparing to 2431.04 seconds of SAM1 and 3623.10 seconds of SAM2. Also note that MBB outputs an optimal solution while SAM cannot guarantee optimality. When the problem becomes more difficult (i.e. $R^1$ increases), SAM’s average error increases from 0.45% to 1.12% in
Table 5.4: Solving Dynamic Interdiction Problems on IEEE BUS 300 Systems. Second round attack budget is $R^2 = 2$.

<table>
<thead>
<tr>
<th>Budget Str.#</th>
<th>5 $4 \times 10^{10}$</th>
<th>6 $2 \times 10^{12}$</th>
<th>7 $1 \times 10^{14}$</th>
<th>8 $4 \times 10^{15}$</th>
<th>9 $6 \times 10^{17}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBB</td>
<td>4.15</td>
<td>5.34</td>
<td>9.17</td>
<td>26.81</td>
<td>32.52</td>
</tr>
<tr>
<td>SAM1</td>
<td>808.49</td>
<td>890.44</td>
<td>942.88</td>
<td>1836.29</td>
<td>2431.04</td>
</tr>
<tr>
<td>err</td>
<td>0.45%</td>
<td>0.69%</td>
<td>0.81%</td>
<td>1.02%</td>
<td>1.12%</td>
</tr>
<tr>
<td>std</td>
<td>0.50%</td>
<td>0.72%</td>
<td>0.82%</td>
<td>1.04%</td>
<td>1.16%</td>
</tr>
<tr>
<td>SAM2</td>
<td>1438.23</td>
<td>1572.46</td>
<td>1649.00</td>
<td>2730.17</td>
<td>3623.10</td>
</tr>
<tr>
<td>err</td>
<td>0.21%</td>
<td>0.24%</td>
<td>0.35%</td>
<td>0.40%</td>
<td>0.52%</td>
</tr>
<tr>
<td>std</td>
<td>0.26%</td>
<td>0.28%</td>
<td>0.37%</td>
<td>0.47%</td>
<td>0.53%</td>
</tr>
</tbody>
</table>

SAM1 and from 0.21% to 0.52% in SAM2.
Chapter 6

Dynamic Network Interdiction Games with Nested Information

In the multi-stage interdiction problems studied in chapter 5, both the attacker and the defender (the network user) have perfect information about the network state. In some situations, the attacker has only statistical information about the network state, referred as partial information. For instance, the attacker can not observe the results of all attacks, or the attacker may be uncertain about the existence of arcs. In these problems, the attacker may gather information from observing the actions of the defender. This raises the possibility for the defender to deny information to the attacker.

From a game-theoretic approach, these problems can be formulated as zero-sum games with nested information in which one player (the defender) has more information than the other player (the attacker). To find the equilibrium behavior strategies, one can adopt the Linear Programming (LP) formulation developed in (von Stengel, 1996). Since the size of the LP problem grows exponentially with the number of interactions, we develop a method, which exploits the nested information structure and decomposes the multi-stage game into a sequence of one-stage subgames. The method computes the expected value of each subgame as a function of its initial distribution. We show that the equilibrium strategies in the original game can be found by solving these subgames.
6.1 Problem Formulation

In this section, we first formulate an extensive form game with nested information. Then we map this game to the dynamic network interdiction problem where the attacker has imperfect information.

Consider a dynamic zero-sum game with finite stages $T$ ($T \geq 2$) where player $X$ and player $Y$ act by turns. Let $a^t, d^t, t = 1, \ldots, T$ be the player $X$'s action and player $Y$'s action at stage $t$ respectively. Let $s^t$ be an underlying state in the game at stage $t$, which evolves according to the previous states $s^{t-1}$ and players' actions $a^\tau, d^\tau$, for any $\tau \leq t$ with transition probabilities that known by both players. We introduce a third player (called Nature) to represent the evolution of the underlying state, whose action can be denoted as $s^t$, meaning Nature chooses the state $s^t$. Since Nature's strategy is known and its payoff is not considered, this game is still zero-sum for player $X$ and player $Y$. A typical play in this game has the following sequence

$$\sigma^T := [s^0, a^1, d^1, s^1, \ldots, a^T, d^T, s^T]. \quad (6.1)$$

Then the payoff associated with that play is determined by the sequence of underlying states and players' actions. As the game is zero-sum, when the game ends with a play $\sigma^T$, let $c_{\sigma^T}$ be player $X$'s payoff, and $-c_{\sigma^T}$ be player $Y$'s payoff.

Define a special class of sequences $\sigma^t$ as the history of the game

$$\sigma^t := [s^0, a^1, d^1, s^1, \ldots, a^t, d^t, s^t],$$

which includes all the actions up to time $t$ from the beginning. We assume both players have perfect recall, i.e., each of them knows what it knew and what it did before. Assume player $X$ has imperfect information about the game, i.e., it does not know the underlying state $s^t, t = 0, \ldots, T$. However it sees all past actions by Player
Y, and remembers its own past actions. Then player X's information set right before
taking action $a^t$ is

$$I^{X,t} := [a^1, d^1, \ldots, a^{t-1}, d^{t-1}].$$

Assume player Y has perfect information about the game. Its information set right
before taking action $d^t$ is

$$I^{Y,t} := [s^0, a^1, d^1, s^1, \ldots, a^{t-1}, d^{t-1}, s^{t-1}, a^t].$$

Assume the evolution of $s^t$ depends on all previous actions and underlying states,
then Nature's information set right before $s^t$ is

$$I^{Z,t} := [s^0, a^1, d^1, s^1, \ldots, a^{t-1}, d^{t-1}, s^{t-1}, a^t, d^t].$$

In the extensive form, the sequence of actions in this model at each stage is as follows:
Nature selects state $s^{t-1}$, then Player X selects $a^t$, followed by Player Y selecting
$d^t$, and Nature subsequently selecting $s^t$. Each sequence corresponds to a node in
the game tree, which is represented by a string of actions reaching the node. Let's
extend the concept of sequence to include the substrings of actions of a node (called
subsequence). Then in perfect recall games, each information set can be represented
by a common subsequence of the nodes within that set, as the above $I^{n,t}$ for $n =
X, Y, Z$.

**Definition 6.1.1** A dynamic zero-sum game is said to have **nested information**
if and only if the information set of one player (the less-informed player) is a subset
of the information set of the other player (the more-informed player) at any stage.

Since $I^{X,t} \subset I^{Y,t}$ for any $t$, therefore the extensive form game defined above has nested
information.

We want to find a set of saddle point strategies of the game, which is a Nash
equilibrium. For general games, the pure strategy equilibrium does not necessarily exist. By Theorem 2.4.9, there is always an equilibrium in terms of mixed strategies. Notice all players in this extensive form game have perfect recall, then according to Theorem 2.4.12, we can find a set of behavior strategies which is realization equivalent to a set of equilibrium mixed strategies.

For any \( n = X, Y, Z \), let \( A(I^n_t) \) be the set of actions at information set \( I^n_t \) and \( S \) be the set of underlying states, all of them are assumed to be finite. Denote player X's behavior strategy as

\[
u := \{u_{IX,t,a_t} | \forall t = 1, \ldots, T, \forall I^{X,t}, \forall a_t \in A(I^{X,t})\},
\]

with \( u_{IX,t,a_t} \) as the probability of taking action \( a_t \) at \( I^{X,t} \). Similarly, player Y's behavior strategy is

\[
u := \{v_{IY,t,d_t} | \forall t = 1, \ldots, T, \forall I^{Y,t}, \forall d_t \in A(I^{Y,t})\},
\]

with \( v_{IY,t,d_t} \) as the probability of taking action \( d_t \) at \( I^{Y,t} \). Denote Nature's strategy as

\[
u := \{r_{IZ,t,s_t} | \forall t = 1, \ldots, T, \forall I^{Z,t}, \forall s_t \in A(I^{Z,t})\},
\]

with \( r_{IZ,t,s_t} := Prob\{s|I^{Z,t}\} \) as the transition probability of \( s_t \) at \( I^{Z,t} \), which is known by all players. Nature also selects the initial underlying state \( s^0 \) by the following probability distribution

\[
u^0 := \{\beta^0_s | \forall s \in S\},
\]

where \( \beta^0_s := Prob\{s\} \) is the probability of Nature selects \( s \). Given \( \beta^0 \) and strategies \( u, v, r \), the realization probability for a history \( \sigma^\tau, \forall \tau = 0, \ldots, T \) is

\[
Prob\{\sigma^\tau | u, v, r, \beta^0\} := \beta^0_s \prod_{t=1}^{\tau} u_{IX,t,a_t} v_{IY,t,d_t} r_{IZ,t,s_t}.
\] (6.2)
where \( s^0, I^{X,t}, a^t, I^{Y,t}, d^t, I^{Z,t}, s^t \) are all on the path of the history \( \sigma^T \). Summing up all plays with their realization probabilities, the expected payoff to player \( X \) is

\[
\sum_{\sigma^T} \beta^0 \prod_{t=1}^{T} u_{I^{X,t}, a^t} v_{I^{Y,t}, d^t} r_{I^{Z,t}, s^t} c_{a^t}.
\] (6.3)

The dynamic network interdiction problem where the attacker has imperfect information as described before can be mapped into the above extensive form game. The underlying state \( s^t (t = 0, \ldots, T) \) is the network state after the \( t \)-th attack, referred as the availability of arcs at time \( t \). The attacker is player \( X \), with \( a^t \) as the set of arcs that are attacked at stage \( t \). The attacked arc may exist; if it exists, the defender (as player \( Y \)) must decide whether to send flows on it (as action \( d^t \)) after observing the attacker’s action. If the defender refrains from using existing arcs, its deception cost is proportional to the unused network capacity. The attacker does not know the exact network state \( s^0 \) but has its statistical information, i.e., the attacker knows the initial distribution of \( \beta^0 \). The attacker can not see the result of interdiction \( s^t \). However it can observe the defender’s actions \( d^t \) and update its statistical information. Based on that, the attacker adjusts its consequent strategy. When the game ends at a play \( \sigma^T \), the attacker’s payoff (as \( c_{a^T} \)) is the sum of defender’s deception costs in all periods minus the maximum flow in the network after all the attacks. The sequence of actions in the dynamic network interdiction problem is shown in Fig. 6.1.

To facilitate the following discussion, let’s introduce some concepts and notation.

**Definition 6.1.2** Given a sequence of \( \sigma^T = [s^0, a^1, d^1, s^1, \ldots, a^T, d^T, s^T] \), define its projections on subspaces (represented by their typical elements) \( s^t, a^t, d^t \) and \( I^{X,t}, I^{Y,t} \).
Figure 6.2: Realization plans linearize the realization probability of a play $\sigma^T$.

$I^{Z,t}, \sigma^t$ as

$s^t(\tilde{\sigma}^T) := \tilde{s}^t$; $a^t(\tilde{\sigma}^T) := \tilde{a}^t$; $d^t(\tilde{\sigma}^T) := \tilde{d}^t$;

$\sigma^t(\tilde{\sigma}^T) := [s^0, \tilde{a}^1, \tilde{d}^1, \tilde{s}^1, \ldots, \tilde{a}^t, \tilde{d}^t, s^t]$;

$I^{X,t}(\tilde{\sigma}^T) := [\tilde{a}^1, \tilde{d}^1, \ldots, \tilde{a}^{t-1}, \tilde{d}^{t-1}]$;

$I^{Y,t}(\tilde{\sigma}^T) := [s^0, \tilde{a}^1, \tilde{d}^1, \ldots, \tilde{a}^{t-1}, \tilde{d}^{t-1}, \tilde{s}^{t-1}, \tilde{a}^t]$;

$I^{Z,t}(\tilde{\sigma}^T) := [s^0, \tilde{a}^1, \tilde{d}^1, \ldots, \tilde{a}^{t-1}, \tilde{d}^{t-1}, \tilde{s}^{t-1}, \tilde{a}^t, \tilde{d}^t]$.

Similarly, for a sequence of $\tilde{I}^{Y,\tau} = [\tilde{s}^0, \tilde{a}^1, \tilde{d}^1, \ldots, \tilde{a}^{\tau-1}, \tilde{d}^{\tau-1}, \tilde{s}^{\tau-1}, \tilde{a}^\tau]$, define its projection on $I^{X,t}, s^t, a^t, d^t$ for any $t \leq \tau$ as

$I^{Y,t}(\tilde{I}^{Y,\tau}) := [s^0, \tilde{a}^1, \tilde{d}^1, \ldots, \tilde{a}^{t-1}, \tilde{d}^{t-1}, \tilde{s}^{t-1}, \tilde{a}^t]$;

$I^{X,t}(\tilde{I}^{Y,\tau}) := [\tilde{a}^1, \tilde{d}^1, \ldots, \tilde{a}^{t-1}, \tilde{d}^{t-1}], s^t(\tilde{I}^{Y,\tau}) = \tilde{s}^t$, $a^t(\tilde{I}^{Y,\tau}) = \tilde{a}^t$, $d^t(\tilde{I}^{Y,\tau}) = \tilde{d}^t$.

For a sequence of $\tilde{I}^{X,\tau} = [\tilde{a}^1, \tilde{d}^1, \ldots, \tilde{a}^{\tau-1}, \tilde{d}^{\tau-1}]$, define its projection on $I^{X,t}, s^t, a^t, d^t$ for any $t \leq \tau$ as

$I^{Y,t}(\tilde{I}^{X,\tau}) := [s^0, \tilde{a}^1, \tilde{d}^1, \ldots, \tilde{a}^{t-1}, \tilde{d}^{t-1}, \tilde{s}^{t-1}, \tilde{a}^t]$;

$I^{X,t}(\tilde{I}^{X,\tau}) := [\tilde{a}^1, \tilde{d}^1, \ldots, \tilde{a}^{t-1}, \tilde{d}^{t-1}], s^t(\tilde{I}^{X,\tau}) = \tilde{s}^t$, $a^t(\tilde{I}^{X,\tau}) = \tilde{a}^t$, $d^t(\tilde{I}^{X,\tau}) = \tilde{d}^t$.

Definition 6.1.3 If a shorter sequence $\tilde{I}_s$ is the corresponding projection of a longer sequence $\tilde{I}_t$, i.e., $I_s(\tilde{I}_t) = \tilde{I}_s$, then $\tilde{I}_t$ is said to be a successor of $\tilde{I}_s$, denoted as $\tilde{I}_t \succ \tilde{I}_s$ or $\tilde{I}_s \prec \tilde{I}_t$.

Notice (6.3) is a complex function of $u$ and $v$. To formulate the objective function in a bilinear form, we adapt the sequence form in (von Stengel, 1996). The idea is to linearize the realization probability of plays by replacing the behavior strategies with the realization plans, as shown in Fig. 6.2.
For a history \( \sigma^T \) for any \( t = 1, \ldots, T \), define \( r_{\sigma^T} := \prod_{t=1}^T r_{I^X_t, a_t, \sigma^T} \) as Nature's cumulative transition probability. According to von Stengel's sequence formulation, we can separate the realization probability of a play into three parts: Nature's cumulative transition probability \( r_{\sigma^T} \), player \( X \)'s realization plan

\[
x := \{ x_{I^X_t, a_t} | \forall t = 1, \ldots, T, \forall I^X_t, a_t \in A(I^X_t) \},
\]

and player \( Y \)'s realization plan (with initial distribution \( \beta^0 \)).

\[
y := \{ y_{I^Y_t, d_t} | \forall t = 1, \ldots, T, \forall I^Y_t, d_t \}
\]

where \( x_{I^X_t, a_t}, y_{I^Y_t, d_t} \) can be defined recursively by

\[
x_{I^X_t, a_t} = u_{I^X_t, a_t}, \quad \forall a_t;
\]

\[
x_{I^X_t, a_t} = x_{I^X_{t-1}(I^X_t), a_{t-1}(I^X_t)} u_{I^X_{t-1}(I^X_t), a_{t-1}(I^X_t)}, \quad \forall t > 1, I^X_t, a_t;
\]

\[
y_{I^Y_t, d_t} = \beta_{\sigma^{I^Y_t, d_t}}^0, \quad \forall I^Y_t, d_t,
\]

\[
y_{I^Y_t, d_t} = y_{I^Y_{t-1}(I^Y_t), d_{t-1}(I^Y_t)} v_{I^Y_{t-1}(I^Y_t), d_{t-1}(I^Y_t)}, \quad \forall t > 1, I^Y_t, d_t.
\]

Then the constraints on \( x, y \) are

\[
\sum_{a_t} x_{I^X_t, a_t} = 1; \quad \sum_{a_t} x_{I^X_t, a_t} = x_{I^X_{t-1}(I^X_t), a_{t-1}(I^X_t)}, \quad \forall t > 1, I^X_t; \quad
\]

\[
\sum_{d_t} y_{I^Y_t, d_t} = \beta_{\sigma^{I^Y_t, d_t}}^0, \quad \forall I^Y_t; \quad \sum_{d_t} y_{I^Y_t, d_t} = y_{I^Y_{t-1}(I^Y_t), d_{t-1}(I^Y_t)}, \quad \forall t > 1, I^Y_t.
\]

Let \( X \) be the feasible set of \( x \) and \( Y(\beta^0) \) be the feasible set of \( y \), representing the above constraints on \( x, y \). \( X, Y(\beta^0) \) can be written concisely as

\[
X := \{ x \geq 0 | Ex = e \}; \quad Y(\beta^0) := \{ y \geq 0 | Fy = B\beta^0 \}
\]

(6.4)

where \( E, e, F \) and \( B \) are matrices (or vectors) derived from the coefficients of the
constraint equations. Specifically, let \(|I^X|, |I^Y|\) be the total numbers of information sets \(I^{X,t}, I^{Y,t}\) respectively and \(|A^X|, |A^Y|\) be the total numbers of actions across all of player X's information sets and player Y's all information sets, respectively. \(E\) is a \(|I^X| \times |A^X|\) matrix and \(F\) is a \(|I^Y| \times |A^Y|\) matrix. \(e\) is a \(|I^X|\)-dimension vector with \(e_{I^X,1} = 1\) and all other elements being 0. \(B\) is a \(|I^Y| \times |S|\) dimension sparse matrix.

By Proposition 2.4.14, \(x, y\) are realization equivalent to the behavior strategies of \(u, v\) (with \(\beta^0\)) where they are derived from. Then we can take \(x, y\) as players' strategies. The realization probability is equal to

\[
Prob\{\sigma^T | x, y\} := x_I x_T (\sigma^T, a_T(\sigma^T)) y_{I^Y, T(\sigma^T), s_T(a_T)} c_T(\sigma^T), r_T, \sigma^T.
\]

Remember \(c_{\sigma^T}\) is defined as the payoff at the play \(\sigma^T\). Therefore the expected payoff under strategies \(x, y\) is

\[
J(x, y) := \sum_{\sigma^T} x_I x_T (\sigma^T, a_T(\sigma^T)) y_{I^Y, T(\sigma^T), s_T(a_T)} r_T c_{\sigma^T} := x^T C y, \tag{6.5}
\]

where \(C\) is a sparse matrix with non-zero elements derived from \(c_{\sigma^T r_{\sigma^T}}\) for all \(\sigma^T\). Any non-zero element in \(C\) corresponding to \(\sigma^T\) has the row index associated with \(x_I x_T (\sigma^T, a_T(\sigma^T))\) and the column index associated with \(y_{I^Y, T(\sigma^T), s_T(a_T)}\). Since \(C\) are known constants, then \(J(x, y)\) is a bilinear function on \(x, y\). In this game, player X's objective is to maximize \(J(x, y)\) by controlling \(x\) and player Y seeks to minimize \(J(x, y)\) by controlling \(y\). Therefore the problem of finding equilibrium strategies can be formulated as the following optimization problems

\[
\max_{x \in X} \min_{y \in Y(\beta^0)} J(x, y), \tag{6.6}
\]

\[
\min_{y \in Y(\beta^0)} \max_{x \in X} J(x, y). \tag{6.7}
\]
Consider the inner maximization problem in (6.7)

\[
\begin{align*}
\max_x & \quad x'Cy \\
\text{s.t.} & \quad Ex = e; x \geq 0.
\end{align*}
\]

Let \( p \) be the dual variable corresponding to the constraint \( Ex = e \), then the dual problem is

\[
\begin{align*}
\min_p & \quad e'p \\
\text{s.t.} & \quad E'p \geq Cy.
\end{align*}
\]

Replacing the inner maximization problem in (6.7) with its dual, we have

\[
\min_{y,p} e'p \quad \text{s.t.} \quad Fy = B\beta^0, y \geq 0, E'p \geq Cy.
\]

(6.8)

**Lemma 6.1.4** Problem (6.7) is equivalent to (6.8), i.e, they have the same optimal value and the same optimal \( y \).

**Proof:** Notice the inner maximization problem in (6.7) is feasible for any \( y \in Y(\beta^0) \) and \( C, x, y \) are all bounded, then its objective value is also bounded. By Strong Duality of the LP problem, its dual problem is feasible, bounded, and has the same optimal value as the primal. Therefore, the optimal solution and the optimal value in (6.8) are the same as in (6.7).

Consider the inner minimization problem in (6.6)

\[
\begin{align*}
\min_y & \quad x'Cy \\
\text{s.t.} & \quad Fy = B\beta^0; y \geq 0.
\end{align*}
\]

Let \( q \) be the dual variable corresponding to the constraint \( Fy = B\beta^0 \), then the dual
problem is
\[
\begin{aligned}
\max_q \beta^0 B'q \\
\text{s.t.} \quad F'q \leq C'x.
\end{aligned}
\]
Replacing the inner minimization problem in (6.6) with its dual, we have
\[
\max_{x,q} \beta^0 B'q \quad \text{s.t.} \quad Ex = e; x \geq 0; F'q \leq C'x.
\]

Lemma 6.1.5 Problem (6.6) is equivalent to (6.9), i.e., they have the same optimal value and the same optimal $x$.

Proof: Notice the inner minimization problem in (6.6) is feasible for any $x \in X$ and $C$, $x$, $y$ are all bounded, then its objective value is also bounded. By Strong Duality of the LP problem, its dual is feasible, bounded, and has the same optimal value as the primal. Therefore, the optimal solution $x^*$ and the optimal value of (6.6) are the same as (6.7).

By taking the dual of (6.8) with $x$ corresponds to $E'p \geq Cy$ and $q$ corresponds to $Fy = B\beta^0$, one recovers the formulation of (6.9). Because both problems are bounded and feasible, Strong Duality still holds. As a result, (6.6) and (6.7) have the same optimal value. As a simple application of more general minimax theorems (Fan, 1953), we have

Theorem 6.1.6 Problem (6.6) and (6.7) have the same optimal value, therefore the optimal solutions of the outer problems in (6.9) and (6.8) are the saddle-point strategies.

Proof: Let $x^*, y^*$ be the optimal solutions of the outer problems in (6.9) and (6.8). By definition,
\[
\min_{y \in Y(\beta)} x^*Cy \leq x^*Cy^* \leq \max_{x \in X} x'Cy^*.
\]
On the other side, from previous discussion, we have problem (6.6) and (6.7) have the same optimal value, i.e.,

$$\min_{y \in Y(\beta^0)} x^*Cy = \max_{x \in X} x^*Cy^*.$$ 

Therefore $\langle x^*, y^* \rangle$ are saddle-point strategies.

Denote the common optimal value of (6.6) and (6.7) as $V(\beta^0)$. Next we are going to show that $V(\beta^0)$ can be represented by a finite set of support vectors. Before that we first need to show some properties about the feasible set in problem (6.7).

**Lemma 6.1.7** Consider the constraint $F'q \leq C'x$ in problem (6.9), if $F'q = 0$, then $q = 0$.

**Proof:** We need to investigate the details of $F'q = 0$. The inner minimization over $q$ in problem (6.9) was derived from the dual of

$$\left\{ \begin{array}{l}
\min_{y \geq 0} x^*Cy \\
s.t. \quad Fy = B \beta^0.
\end{array} \right.$$ 

Explicitly, the constraints of $Fy = B \beta^0$ are

$$\sum_{d^t} y_{d^t Y,Y_1} = \beta^0_{\delta Y_1 Y_1}, \forall Y_1,$$

$$\sum_{d^t} y_{d^t Y,t} = \delta_{(d^t Y,Y_t), (d^t-1 Y_{t-1})}, \forall t > 1, Y_t.$$ 

Let $q_{d^t Y,Y_1}$ be the dual variable for constraint $\sum_{d^t} y_{d^t Y,Y_1} = \beta^0_{\delta Y_1 Y_1}$ and $q_{d^t Y,t}$ be the dual variable for constraint $\sum_{d^t} y_{d^t Y,t} = \delta_{(d^t Y,Y_t), (d^t-1 Y_{t-1})}$ for $t > 1$. Denote the coefficient corresponding to $y_{d^t Y,t}$ in $x^*Cy$ as $c_{d^t Y,t}$, then the Lagrange function of
the above dual problem is

\[
\sum_{t=1}^{T} \sum_{v,t} c_{Y,t} y_{Y,t} + \sum_{t=1}^{T} \left( \beta_{0}^{0} y_{Y,t} - \sum_{d} y_{Y^2,t} \right) q_{IY,1}
\]

\[+ \sum_{t=2}^{T} \sum_{v,t} (y_{Y,t-1} y_{Y,t-1} - \sum_{d} y_{Y,t-1,d} q_{IY,t})q_{IY,t} \]

\[= \sum_{t=1}^{T} \sum_{v,t} \beta_{0}^{0} y_{Y,t} q_{IY,1} + \sum_{t=2}^{T} \left( c_{Y,t} y_{Y,t} - q_{IY,t-1} + \sum_{j} q_{IY,t} y_{IY,t-1,d} \right) q_{IY,t} \]

\[+ \sum_{t=2}^{T} \left( c_{Y,t} - q_{IY,t} \right) y_{IY,t} \cdot d_{t}. \]

Notice \( y \geq 0 \), the dual constraint \( F'q \leq C'x \) can be written explicitly as

\[q_{IY,t-1} - \sum_{IY,t-1,d} q_{IY,t} \leq c_{Y,t-1,d} q_{IY,t-1}, \forall t = 2, \ldots, T, IY,t-1, d_{t-1}; \]

\[q_{IY,t} \leq c_{Y,t} q_{IY,t}, \forall IY,t, d_{t}. \]

Then the details of \( F'q = 0 \) are

\[q_{IY,t} = 0, \forall IY,t; \quad q_{IY,t-1} - \sum_{IY,t-1,d} q_{IY,t} = 0, \forall t \leq T, IY,t-1, d_{t-1} \]

Since \( q_{IY,t} = 0 \) and we can have \( q_{IY,t-1} = 0 \) based on \( q_{IY,t} = 0 \) and \( q_{IY,t-1} - \sum_{IY,t-1,d} q_{IY,t} = 0 \). Therefore all elements of \( q \) are equal to 0. \( \blacksquare \)

**Theorem 6.1.8** There is a finite set of vectors \( Q \), such that

\[V(\beta) = \max_{q \in Q} \beta'q, \quad \forall \beta \in [0, 1]^{[y]}. \]

**Proof:** By the definition of \( V(\beta) \), we just need to show that the optimal value of problem (6.9) can be represented by \( Q \). By lemma 6.1.6, the rows of \( F \) are linearly independent, then the feasible set of the problem (6.9) contains at least an extreme point (Theorem 2.6 in (Bertsimas and Tsitsiklis, 1997)). Notice \( x, y, C \)
are all bounded, then the saddle point value of \( V(\beta) \) for any \( \beta \in [0, 1]^{[S]} \) must be bounded. Then for any \( \beta \), there must exist an extreme point that is an optimal solution (Theorem 2.7 in (Bertsimas and Tsitsiklis, 1997)). Therefore

\[
V(\beta) = \max_{(x,a) \in \{n^i\}_{i=1}^I} \beta' B' q,
\]

where \( \{n^i\}_{i=1}^I \) is the set of extreme points of the feasible set in (6.9). Denote the projection of \( \{n^i\}_{i=1}^I \) on the subspace of \( q \) as \( \hat{Q} := \{\hat{q}_i\}_{i=1}^I \). Define \( Q := \{q_i | q_i = B' \hat{q}_i, \forall \hat{q}_i \in \hat{Q}\} \), then we have

\[
V(\beta) = \max_{\hat{q} \in \hat{Q}} \beta' B' \hat{q} = \max_{q \in Q} \beta' q.
\]

6.2 Subgame Decomposition Method

One can obtain saddle-point strategies by solving LP problems (6.8) and (6.9). However, their sizes grow exponentially with the number of stages, which limits the applications to large games. Here we propose a method, which decomposes the multi-stage game into a sequence of one stage subgames based on its nested information structure. By solving those subgames sequentially, we can obtain the equilibrium strategies of the original game.

6.2.1 Subgame Equilibrium

In a perfect recall game, because any player can remember what it did and what it knew before, any nodes within an information set of a player can not come from different information sets of the same player. Also notice that for games with nested information, any information set of the more-informed player can not come from different information sets of the less-informed player at the same level. Therefore
any two information sets of the less-informed player at the same level can not have common successors.

**Definition 6.2.1** In a perfect recall game, given an information set of the less-informed player $I^{X,t}$, $t < T$ and a probability distribution on that set $\beta^{I^{X,t}}$, a subgame from $I^{X,t}$ with $\beta^{I^{X,t}}$, denoted as $G(I^{X,t}, \beta^{I^{X,t}})$, is an extensive form game inheriting $I^{X,t}$ and all its successors in the original game tree, such that, the subgame starts from Nature selecting nodes in $I^{X,t}$ with the probability distribution $\beta^{I^{X,t}}$.

We will decompose the multi-stage game into a set of single stage games, with payoffs from subgames that have one less stage. Note that with the same procedure, one can further decompose these subgames, until the lowest level game contains only one information set of the less-informed player. The idea of the subgame decomposition method is shown in Fig. 6-3. According to (6.2) when $t = 1$, the players’ first strategies control the realization probability of the history $\sigma^1$, which are in multiple information sets $I^{X,2}$. Therefore given the players’ first strategies, the distributions on these information sets $I^{X,2}$ are fixed. Then we can decompose the original game into subgames, each of them starts at one $I^{X,2}$ with the initial distribution $\beta^{I^{X,2}}$ derived from the players’ first strategies. On the other side, once the payoff functions of all the subgames $G(I^{X,t}, \beta^{I^{X,t}})$ are given, one can find the equilibrium first strategies of the original game.
To decompose the original game, we separate the players' first strategies from their remaining strategies. For any play $\sigma^T$, the realization probability is

$$\beta_s^0 u_{I^X, a_1} v_{I^Y, d_1} r_{I^Z, s_1} \prod_{t=2}^T u_{I^X, a_t} v_{I^Y, d_t} r_{I^Z, s_t}$$

where $I^X_t, a^t, I^Y_t, d^t, I^Z_t, s^t, t = 1, \ldots, T$ are all on the path of $\sigma^T$. Define the first interaction strategies before $I^{X,1}$ as

$$x^1 := \{u_{a^1} | u_{a^1} := u_{I^X, a^1} \forall a^1\},$$

$$y^1 := \{y_{I^Y, d^1} | y_{I^Y, d^1} := \beta_s^0 v_{I^Y, d^1} \forall I^Y, d^1\}.$$ 

And define the remaining strategies after $I^{X,1}$ as

$$x^r := \{x_{I^X, a^t} | \forall I^{X, t}, a^t, t > 1\}; \quad y^r := \{y_{I^Y, d^t} | \forall I^{Y, t}, d^t, t > 1\}$$

with

$$x_{I^X, a^t} := u_{I^X, a^t} \prod_{\tau=2}^{t-1} u_{I^X, a^{\tau}(I^{X, \tau}), a^{\tau}(I^{X, \tau})}, \forall I^{X, t}, a^t;$$

$$y_{I^Y, d^t} := y_{I^Y, d^t} \prod_{\tau=2}^{t-1} v_{I^Y, d^{\tau}(I^{Y, \tau}), d^{\tau}(I^{Y, \tau})}, \forall I^{Y, t}, t \geq 2.$$ 

Based on their definitions, the probability constraints on $x^1, y^1, x^r, y^r$ are

$$\sum_{a^1} x^1_{a^1} = 1; \quad \sum_{d^1} y^1_{I^Y, d^1} = \beta_s^0 v_{I^Y, d^1}, \forall I^Y, d^1;$$

$$\sum_{d^2} y^1_{I^Y, d^2} = y^1_{I^Y, d^2}, \forall I^{Y, 2}; \quad \sum_{a^2} x^r_{I^{X, 2}, a^2} = 1, \forall I^{X, 2};$$

$$\sum_{a^t} x^r_{I^X, a^t} = x^r_{I^X, a^{t-1}(I^{X, t}), a^{t-1}(I^{X, t})}, \forall I^{X, t}, t > 2;$$

$$\sum_{d^t} y^r_{I^Y, d^t} = y^r_{I^Y, d^t}, \forall I^{Y, t}, t > 2.$$
Denote the feasible sets for $x_1, x^r$ as $X_1, X^r$ respectively. Notice the constraints of $y^r$ depend on $y^1$ and the constraints of $y^1$ depend on $\beta^0$, denote the feasible set of $y^r$ as $Y^r(y^1)$ and the feasible set of $y^1$ as $Y^1(\beta^0)$.

Denote $r_{\sigma T} := \prod_{\tau = 2}^{T} r_{I^Z T(\sigma T), \sigma (\sigma T)}$, then the realization probability of $\sigma^T$ can be written as

$$\sum_{\sigma T} x_{a_1(\sigma T)} I^{Z, 1}(\sigma T), s_1(\sigma T) x^r_{I^X T(\sigma T), a_T(\sigma T)} y^r_{I^Y T(\sigma T), d_T(\sigma T)} r_{\sigma T} c_{\sigma T} \alpha_{\sigma T}.$$

Given $x^1, y^1, x^r, y^r$ and $\beta^0$, the expected payoff is

$$\tilde{J}(x^1, y^1, x^r, y^r) := \sum_{\sigma T} x_{a_1(\sigma T)} I^{Z, 1}(\sigma T), s_1(\sigma T) x^r_{I^X T(\sigma T), a_T(\sigma T)} y^r_{I^Y T(\sigma T), d_T(\sigma T)} r_{\sigma T} c_{\sigma T};$$

where the second sum groups plays $\sigma^T$ into the different information sets $I^{X, 2}$ that they belong to. By the definition of $x^1, x^r$, maximizing $x$ in $J(x, y)$ is equivalent to maximizing $x^1, x^r$ in $\tilde{J}(x^1, y^1, x^r, y^r)$. Similarly by the definition of $y^1, y^r$, minimizing $y$ in $J(x, y)$ is equivalent to minimizing $y^1, y^r$ in $\tilde{J}(x^1, y^1, x^r, y^r)$, therefore (6.6) and (6.7) are equal to

$$J_{MM} := \max_{x^1 \in X^1, x^r \in X^r, y^1 \in Y^1(\beta^0), y^r \in Y^r(y^1)} \\min \tilde{J}(x^1, y^1, x^r, y^r);$$

$$J_{mm} := \min_{y^1 \in Y^1(\beta^0), y^r \in Y^r(y^1)} \\max x^1 \in X^1, x^r \in X^r \tilde{J}(x^1, y^1, x^r, y^r).$$

Notice $V(\beta^0)$ is defined as the optimal value of (6.6) and (6.7), then we have

$$J_{MM} = J_{mm} = V(\beta^0).$$
Consider the following optimization problems

\[
J_{mmMM} := \min_{y^1 \in Y^1(\beta^0)} \max_{x^1 \in X^1} \min_{y^r \in Y^r(y^1)} \max_{x^r \in X^r} \tilde{J}(x^1, y^1, x^r, y^r); \quad (6.12)
\]

\[
J_{MmMM} := \max_{x^1 \in X^1} \min_{y^1 \in Y^1(\beta^0)} \max_{x^r \in X^r} \min_{y^r \in Y^r(y^1)} \tilde{J}(x^1, y^1, x^r, y^r), \quad (6.13)
\]

where (6.12) is the result of exchanging the optimization orders of \(x^1\) and \(y^r\) in (6.11) and (6.13) is the result of exchanging the optimization orders of \(y^1\) and \(x^r\) in (6.10).

We will show that all these problems have the same optimal value.

**Lemma 6.2.2** \(J_{mmMM}\) and \(J_{MmMM}\) are equal to \(V(\beta^0)\).

**Proof:** Switch the optimization orders of \(x^1\) and \(y^r\) in (6.11), by the minimax theorem (Fan, 1953), we have

\[
\min_{y^1 \in Y^1(\beta^0)} \max_{x^1 \in X^1} \min_{y^r \in Y^r(y^1)} \max_{x^r \in X^r} \tilde{J}(x^1, y^1, x^r, y^r) \geq \min_{y^1 \in Y^1(\beta^0)} \max_{x^1 \in X^1} \min_{y^r \in Y^r(y^1)} \max_{x^r \in X^r} \tilde{J}(x^1, y^1, x^r, y^r)
\]

Similarly switch the optimization orders of \(y^1\) and \(x^r\) in (6.10), we have

\[
\max_{x^1 \in X^1} \min_{y^1 \in Y^1(\beta^0)} \max_{x^r \in X^r} \min_{y^r \in Y^r(y^1)} \tilde{J}(x^1, y^1, x^r, y^r) \leq \max_{x^1 \in X^1} \min_{y^1 \in Y^1(\beta^0)} \max_{x^r \in X^r} \min_{y^r \in Y^r(y^1)} \tilde{J}(x^1, y^1, x^r, y^r).
\]

Therefore, we have

\[
J_{mmMM} \leq J_{MmMM} = V(\beta^0) = J_{MMmm} \leq J_{MmMM}.
\]
On the other side, switching the orders of optimization between $x^1, y^1$ and then $x^r, y^r$ in (6.13), by the minimax theorem, we have

$$\max_{x^1 \in X^1} \min_{y^1 \in Y^1(\beta^0)} \max_{x^r \in X^r} \min_{y^r \in Y^r(\beta^0)} J(x^1, y^1, x^r, y^r)$$

$$\leq \min_{y^1 \in Y^1(\beta^0)} \max_{x^1 \in X^1} \min_{x^r \in X^r} \max_{y^r \in Y^r(\beta^0)} J(x^1, y^1, x^r, y^r)$$

Therefore $J_{mMmM} \geq J_{MmMm}$. Combined this result with $J_{mMmM} \leq V(\beta^0) \leq J_{MmMm}$, we have $J_{mMmM} = J_{MmMm} = V(\beta^0)$. $\blacksquare$

**Theorem 6.2.3** The optimal solution $x^{1*}$ of the outer maximization problem in (6.13) and the optimal solution $y^{1*}$ of the outer minimization problem in (6.12) are the saddle point strategies of the first interaction.

**Proof:** By lemma 6.2.2, $J_{mMmM} = V(\beta^0)$, which means player $X$ can not get better payoff by changing $x^1, x^r$ when player $Y$ plays $y^{1*}$. Similarly, $J_{MmMm} = V(\beta^0)$ means player $Y$ can not get better payoff by changing $y^1, y^r$ when player $X$ plays $x^{1*}$. Therefore, $x^{1*}, y^{1*}$ are saddle point strategies for the first interaction. $\blacksquare$

Given $x^1, y^1$, the probability of reaching $I^{X,2}$ for any $I^{X,2}$ is

$$P_{Ix,2}(x^1, y^1) := x^1_{a_1(I^{X,2})} \sum_{\sigma^1 \succ I^{X,2}} y^1_{IY^1(\sigma^1), d^1(\sigma^1), IZ^1(\sigma^1), s^1(\sigma^1)}.$$

For any $P_{Ix,2} > 0$, we can calculate the conditional probability for any $\sigma^1 \succ I^{X,2}$ given the game arrives at $I^{X,2}$

$$\beta^1_{\sigma^1}(x^1, y^1) := x^1_{a_1(\sigma^1)} y^1_{IY^1(\sigma^1), d^1(\sigma^1), IZ^1(\sigma^1), s^1(\sigma^1)} / P_{Ix,2}$$

$$= y^1_{IY^1(\sigma^1), d^1(\sigma^1), IZ^1(\sigma^1), s^1(\sigma^1)} / \sum_{\sigma^1 \succ I^{X,2}} y^1_{IY^1(\sigma^1), d^1(\sigma^1), IZ^1(\sigma^1), s^1(\sigma^1)}.$$

Denote $\beta^{I^{X,2}}(x^1, y^1) := \{\beta^1(x^1, y^1)_{\sigma^1}, |\forall \sigma^1 \succ I^{X,2}\}$, which is a distribution on the nodes within $I^{X,2}$. Define the strategies $x^{I^{X,2}}, y^{I^{X,2}}$ on the subgame $G(I^{X,2}, \beta^{I^{X,2}}(x^1, y^1))$.
as
\[
x^{I^X_2} := \{x^{I^X_2}_{I^X_2,a} | \forall I^{X,t} \succ I^X_2\}, \quad y^{I^X_2} := \{y^{I^X_2}_{I^Y,t,d} | \forall I^{Y,t} \succ I^X_2\},
\]

which satisfy the following constraints
\[
\sum_{a_2} x^{I^X_2}_{I^X_2,a_2} = 1; \quad \sum_{d_2} y^{I^X_2}_{I^Y,t,d_2} = \beta^{I^X_2}(x^1, y^1)_a(I^X_2), \forall I^{Y,t} \succ I^X_2;
\]
\[
\sum_{a_t} x^{I^X_2}_{I^X_2,a_t} = x^{I^X_2}_{I^X_2,a_t-1}(I^X_2), \forall I^{X,t} \succ I^X_2;
\]
\[
\sum_{d_t} y^{I^X_2}_{I^Y,t,d_t} = y^{I^X_2}_{I^Y,t-1}(I^X_2), \forall I^{Y,t} \succ I^X_2.
\]

In the subgame \(G(I^X_2, \beta^{I^X_2}(x^1, y^1))\), given \(x^{I^X_2}, y^{I^X_2}\), the expected payoff is
\[
J^{r}_{I^X_2}(x^{I^X_2}, y^{I^X_2}) := \sum_{\sigma^T \succ I^X_2} x^{I^X_2}_{I^X_2,T(\sigma^T),a^T(\sigma^T)} y^{I^X_2}_{I^Y,T(\sigma^T),d^T(\sigma^T)} r^{T(\sigma^T)} c^{\sigma^T},
\]
\(x^{I^X_2}, y^{I^X_2}\) can be taken as the projection of \(x^r, y^r\) on \(I^{X^2}\) such that for any \(I^{X,t}, I^{Y,t} \succ I^X_2^2\),
\[
x_{I^X_2,t,a^T}^{I^X_2} = x_{I^X_2,t,a^T}, \quad y_{I^Y,t,d^T}^{I^X_2} = y_{I^Y,t,d^T}^{I^X_2}.
\]

Then \(\tilde{J}(x^1, y^1, x^r, y^r)\) can be written as the convex combination of the subgames' expected payoffs
\[
\tilde{J}(x^1, y^1, x^r, y^r) = \sum_{I^X_2} x^{a_1(I^X_2)} \sum_{\sigma^T \succ I^X_2} r^{T(I^X_2,a^T)} x^{I^X_2}_{I^X_2,T(\sigma^T),a^T(\sigma^T)} y^{I^Y,T(\sigma^T),d^T(\sigma^T)} r^{T(\sigma^T)} c^{\sigma^T}
\]
\[
= \sum_{I^X_2} P_{I^X_2}^1(x^1, y^1) \sum_{\sigma^T \succ I^X_2} x^{I^X_2}_{I^X_2,T(\sigma^T),a^T(\sigma^T)} y^{I^Y,T(\sigma^T),d^T(\sigma^T)} r^{T(\sigma^T)} c^{\sigma^T}
\]
\[
= \sum_{I^X_2} P_{I^X_2}^1(x^1, y^1) J^{r}_{I^X_2}(x^{I^X_2}, y^{I^X_2}).
\]
Denote the feasible sets of $x^{lX,2}, y^{lX,2}$ as $X^{lX,2}, Y^{lX,2}(\beta^{lX,2}(x^1, y^1))$, then (6.12) and (6.13) can be written as

$$\max_{x^1 \in X^1} \min_{y^1 \in Y^1(\beta^0)} \sum_{lX,2} P_{lX,2}(x^1, y^1) \max_{x^{lX,2} \in X^{lX,2}, y^{lX,2} \in Y^{lX,2}(\beta^{lX,2}(x^1, y^1))} \min_{x^{lX,2} \in X^{lX,2}} J_{lX,2}(x^{lX,2}, y^{lX,2});$$

$$\min_{y^1 \in Y^1(\beta^0)} \max_{x^1 \in X^1} \sum_{lX,2} P_{lX,2}(x^1, y^1) \min_{y^{lX,2} \in Y^{lX,2}(\beta^{lX,2}(x^1, y^1))} \max_{y^{lX,2} \in Y^{lX,2}} J_{lX,2}(x^{lX,2}, y^{lX,2}).$$

Consider the inner min-max (or max-min) problems of the above problems

$$\max_{x^{lX,2} \in X^{lX,2}, y^{lX,2} \in Y^{lX,2}(\beta^{lX,2}(x^1, y^1))} J_{lX,2}(x^{lX,2}, y^{lX,2});$$

$$\min_{y^{lX,2} \in Y^{lX,2}(\beta^{lX,2}(x^1, y^1))} \max_{x^{lX,2} \in X^{lX,2}} J_{lX,2}(x^{lX,2}, y^{lX,2}),$$

which have the same structures as in problems (6.7) and (6.6) of the original game with $x^{lX,2}, y^{lX,2}, \beta^{lX,2}(x^1, y^1)$ correspond to $x, y, \beta^0$ respectively. In fact, these are the exact formulations for the subgame $G(I^{X,2}, \beta^{lX,2}(x^1, y^1))$. Therefore, theorem 6.1.6 and 6.1.8 apply, problems (6.14) and (6.15) have the same optimal value, denoted as $V_{lX,2}(\beta^{lX,2}(x^1, y^1))$ and there is a finite set of support vectors $Q_{lX,2}$ such that

$$V_{lX,2}(\beta) = \max_{q \in Q_{lX,2}} \beta \cdot q.$$ 

Substitute (6.14), (6.15) with $V_{lX,2}(\beta^{lX,2}(x^1, y^1))$ in (6.12) and (6.13), we have

$$\max_{x^1 \in X^1} \min_{y^1 \in Y^1(\beta^0)} \sum_{lX,2} P_{lX,2}(x^1, y^1)V_{lX,2}(\beta^{lX,2}(x^1, y^1));$$

$$\min_{y^1 \in Y^1(\beta^0)} \max_{x^1 \in X^1} \sum_{lX,2} P_{lX,2}(x^1, y^1)V_{lX,2}(\beta^{lX,2}(x^1, y^1)).$$

Compared with $J_{mmMM}$ and $J_{nmMM}$, (6.16) and (6.17) are smaller, which are easy to solve. But it also raises the following questions

Q1: How can one obtain the set of support vectors $Q_{lX,2}$ for subgame $G(I^{X,2}, \beta^{lX,2})$?
Q2: Notice (6.16) and (6.17) are not simple bilinear optimization problems, how can one solve them?

Q3: How can one obtain the equilibrium strategies for all interactions rather than the first one?

We will answer these questions in the rest of this section, which leads to the development of the Subgame Decomposition Method.

### 6.2.2 Identifying the Set of Support Vectors

Notice the subgame corresponding to the problems of (6.14) and (6.15) is one stage less than the original game. In any subgame \( G(I^{X,t}, \beta^{I^{X,t}}) \) with \( t > 2 \), we can further separate its first interaction strategies from its remaining strategies with the same process, the corresponding optimization problems will be

\[
\begin{align*}
\max_{x^t \in X^t} \min_{y^t \in Y^t(\beta^t)} \sum_{I^{X,t+1} \subseteq I^{X,t}} P_{I^{X,t+1}}(x^t, y^t)V_{I^{X,t+1}}(\beta^{I^{X,t+1}}(x^t, y^t)); \\
\min_{y^t \in Y^t(\beta^t)} \max_{x^t \in X^t} \sum_{I^{X,t+1} \subseteq I^{X,t}} P_{I^{X,t+1}}(x^t, y^t)V_{I^{X,t+1}}(\beta^{I^{X,t+1}}(x^t, y^t))
\end{align*}
\]

(6.18) (6.19)

where \( x^t, y^t, P_{I^{X,t+1}}, \beta^{I^{X,t+1}}(x^t, y^t) \) are defined similarly as those in the original game,

\[
x^t := \{x^t_{I^{X,t}, a^t} | x^t_{I^{X,t}, a^t} := u_{I^{X,t}, a^t} | \forall a^t\};
\]

\[
y^t := \{y^t_{I^{Y,t}, d^t} | y^t_{I^{Y,t}, d^t} := \beta^t_{I^{Y,t-1}}(I^{Y,t})u_{I^{Y,t}, d^t}, \forall I^{Y,t} \supseteq I^{X,t}\};
\]

\[
P_{I^{X,t+1}}(x^t, y^t) := \sum_{\sigma^t \subseteq I^{X,t+1}} y^t_{I^{Y,t}, d^t(\sigma^t)} r^{I^{Z,t}}(\sigma^t, s^t(\sigma^t));
\]

\[
\beta^t_{I^{X,t+1}}(x^t, y^t) := \sum_{\sigma^t \subseteq I^{X,t+1}} y^t_{I^{Y,t}, d^t(\sigma^t)} r^{I^{Z,t}}(\sigma^t, s^t(\sigma^t)).
\]

This process continues until it reaches \( G(I^{X,T}, \beta^{I^{X,T}}) \), where the payoff for any play \( \sigma^T = [\sigma^{T-1}, a^T, d^T, s^T] \) is given by \( c_{\sigma^T} \). We will use a bottom up procedure to estimate the support vectors in \( Q_{I^{X,2}} \). The ideas of this procedure is shown in Fig.
6.4. For bottom subgames $G(I^{x,t}, \cdot)$, $Q^{x,t+1}$ are the payoffs. For other subgames $G(I^{x,t}, \cdot), t < T$, we will show how to estimate $Q^{x,t}$ with given $Q^{x,t+1}$ by solving a set of one-stage problems.

Let $u_{I^{x,t},a^t}$ be player X's behavior strategy on $I^{x,t}$, $v_{I^{y,t},d^t}$ be player Y's behavior strategy on $I^{y,t}$. Given these strategies, the expected payoff in $G(I^{x,t}, \beta^{x,t})$ can be written as

$$J_{I^{x,t}}(u, v) := \sum_{a^{t-1} \in I^{x,t}} \beta_{a^{t-1}}^{t} \sum_{a^t} u_{I^{x,t}, a^t}$$

$$\sum_{d^t} v_{[\sigma^{t-1}, a^t], d^t} \sum_{s^t} r_{[\sigma^{t-1}, a^t, d^t], s^t} c_{[\sigma^{t-1}, a^t, d^t, s^t]}.$$

The probability constraints on $u, v$ are

$$\sum_{a^t} u_{I^{x,t}, a^t} = 1; u \geq 0; \sum_{d^t} v_{[\sigma^{t-1}, a^t], d^t} = 1, \forall \sigma^{t-1}, a^t; v \geq 0,$$

denoted as $u \in U, v \in V$. Notice $J_{I^{x,t}}(u, v)$ is a bilinear function on $u, v$, by theorem 6.1.6, the saddle point strategies of $G(I^{x,t}, \beta^{x,t})$ are the optimal solutions of the
following problems

\[
\max \min_{u \in U} J_{I \times T}(u, v); \quad \min \max_{v \in V} J_{I \times T}(u, v).
\]

Denote their common optimal value as \( V_{I \times T}(\beta^{IX,T}) \). By substituting \( u_{I \times T,aT} \) with \( x_{I \times T,aT} \) and \( u_{I \times T,aT} \beta_{\sigma_T-1}^{IX,T} \) with \( y_{I \times T,dT} \) in \( J_{I \times T}(u, v) \), the resulting objective function has the form of \( J(x, y) \) in (6.5), therefore theorem 6.1.8 applies, and \( V_{I \times T}(\beta^{IX,T}) \) can be represented as the maximum of a finite set of linear functions. Denote the saddle point strategies as \((u^*, v^*)\), substitute them into \( J_{I \times T}(u^*, v^*) \), \( V_{I \times T}(\beta^{IX,T}) \) is

\[
\sum_{\sigma_T-1} \beta^{IX,T}_{\sigma_T-1} \left[ \sum_{aT} u_{I \times T,aT} \sum_{dT} v_{[\sigma_T-1,aT],dT} \sum_{sT} s_{[\sigma_T-1,aT,dT],sT} \right].
\]

Let \( q_{\sigma_T-1} := \sum_{aT} u_{I \times T,aT} \sum_{dT} v_{[\sigma_T-1,aT],dT} \sum_{sT} s_{[\sigma_T-1,aT,dT],sT} \), then the above equation suggests that \( q := \{q_{\sigma_T-1} | \sigma_T-1 \prec I^{X,T} \} \) is a support vector at the value \( \beta^{IX,T} \). Identifying the rest of the support vectors requires an algorithm for determining the extreme points of the dual polytope; later we will discuss an approximate algorithm for determining these support vectors.

Assume that we have identified the sets of support vectors for \( V_{I \times T}(\beta^{IX,T}) \) for any \( I^{X,T} \), then we use \( V_{I \times T}(\beta^{IX,T}) \) to find the set of support vectors for \( V_{I \times T-1}(\beta^{IX,T-1}) \). Following this procedure inductively, we can eventually identify the set of support vectors for \( V(\beta^0) \). The following discussion will focus on how to find the support vectors of \( V_{I \times T}(\beta^{IX,T}) \) given \( V_{I \times T-1}(\beta^{IX,T-1}) \) for all \( I^{X,t+1} \succ I^{X,t} \).

For a subgame \( G(I^{X,t}, \beta^{IX,t}) \) with \( t < T \), denote the sets of support vectors of its lower level subgames \( G(I^{X,t+1}, \beta^{IX,t+1}) \) with \( I^{X,t+1} = [I^{X,t}, a^t, d^t] \) as

\[
Q_{I^{X,t}} := \{q_k\}_{k=1}^{K^{IX,t+1}}, \quad \forall a^t, d^t.
\]
Then any $V_{I^{X,t+1}}(\beta^{I^{X,t+1}})$ can be written as
\[
V_{I^{X,t+1}}(\beta^{I^{X,t+1}}) := \max_{\lambda \in \Lambda^{I^{X,t+1}}} \sum_{k=1}^{K^{I^{X,t+1}}} \lambda_k \beta^{I^{X,t+1}} \cdot q_k,
\]
where $\Lambda^{I^{X,t+1}} := \{\lambda_k | \sum_{k=1}^{K^{I^{X,t+1}}} \lambda_k = 1, \lambda_k \geq 0\}$ is the set of convex combination coefficients. This is now a maximization over a continuous variable $\lambda$ instead of a discrete variable $k$. Let $u_{I^{X,t},a}$ be player $X$'s behavior strategy on $I^{X,t}$, $v_{[\sigma^{-1},a],d}$ be player $Y$'s behavior strategy on $[\sigma^{-1},a]$. In (6.19) and (6.18), substitute
\[
u_{[\sigma],a} = u_{I^{X,t},a}, \quad y_{[\sigma],a} = \beta^{I^{X,t}}_{[\sigma^{-1},a]} v_{[\sigma^{-1},a],d}
\]
into the objective function, we have
\[
J_{I^{X,t}}(u,v) = \sum_{I^{X,t+1}} P_{I^{X,t+1}}(u,v) V_{I^{X,t+1}}(\beta^{I^{X,t+1}}(u,v))
\]
\[
= \sum_{\sigma^{-1} \in I^{X,t}} \beta^{I^{X,t}}_{\sigma^{-1}} \sum_{a} u_{I^{X,t},a} \sum_{d} v_{[\sigma^{-1},a],d} dt
\]
\[
\max_{\lambda \in \Lambda^{I^{X,t+1}}} \sum_{k=1}^{K^{I^{X,t+1}}} \lambda_k \sum_{\sigma} r_{[\sigma^{-1},a],d} \sigma q_k, [\sigma^{-1},a],d, \sigma
\]
The probability constraints on $u, v$ are
\[
\sum_{a} u_{I^{X,t},a} = 1; u \geq 0; \sum_{d} v_{[\sigma^{-1},a],d} dt = 1, \forall [\sigma^{-1},a]; v \geq 0,
\]
denoted as $u \in U, v \in V$. By theorem 6.2.3, the first interaction saddle point strategies of $G(I^{X,t}, \beta^{I^{X,t}})$ are the optimal solutions of the following problems
\[
\max_{u \in U} \min_{v \in V} J_{I^{X,t}}(u,v); \min_{v \in V} \max_{u \in U} J_{I^{X,t}}(u,v). \quad (6.20)
\]
In $\min_{v \in V} \max_{u \in U} J_{I^{X,t}}(u,v)$, one can merge the maximization over $u$ and the
maximization over \( \lambda \). Define \( w_k^{I_{X,t+1}} := u_{I_{X,t+1},a^t} \lambda_k^{I_{X,t+1}} \) for any \( a^t, I_{X,t+1} \) and \( k \), then

\[
\max_{u \in U} \sum_{a^t} \beta^{I_{X,t}} \sum_{a^t} u_{I_{X,t+1},a^t} \sum_{d^t} u_{[\sigma^{t-1},a^t],d^t} K^{I_{X,t+1}}
\]

\[
= \max_{a^t \in \Lambda_{I_{X,t+1}}} \sum_{k=1}^{K^{I_{X,t+1}}} \lambda_k^{I_{X,t+1}} \sum_{s^t} r_{[\sigma^{t-1},a^t,d^t],s^t} q_{k,s^t} [\sigma^{t-1},a^t,d^t,s^t]
\]

where \( W \) is a set of weights of support vectors in all lower level subgames.

\[
W := \{ w \mid \sum_{k=1}^{K^{I_{X,t+1}}} w_k^{I_{X,t+1}} = \sum_{k=1}^{K^{I_{X,t+1}}} w_k^{I_{X,t+1}}, \text{ if } a^t(\tilde{I}_{X,t+1}) = a^t(\tilde{I}_{X,t+1}); \}
\]

\[
\sum_{a^t} \sum_{k=1}^{K^{I_{X,t+1}}} w_k^{I_{X,t+1}} = 1; w \geq 0 \}
\]

By its definition, \( w \) must satisfy \( \sum_{k=1}^{K^{I_{X,t+1}}} w_k^{I_{X,t+1}} = u_{I_{X,t+1},a^t}(I_{X,t+1}) \) for any \( I_{X,t+1} \). Denote the linear constraints in \( W \) as \( Lw = l \) where \( l \) is vector with dimension as the number of equations in \( W \) and \( L \)'s column number is the dimension of \( w \). According to this definition, we can retrieve \( u_{I_{X,t+1},a^t} \) by summing \( w \) over \( I_{X,t+1} \) and we can also retrieve \( \lambda_k^{I_{X,t+1}} \) by normalized \( w \) within \( I_{X,t+1} \). Denote the objective function in the above maximization problem as

\[
\tilde{J}_{I_{X,t}}(w, v) := \sum_{a^t \in \Lambda_{I_{X,t}}} \beta^{I_{X,t}} \sum_{a^t} u_{[\sigma^{t-1},a^t],d^t} \sum_{k=1}^{K^{I_{X,t+1}}} w_k^{I_{X,t+1}} \sum_{s^t} r_{[\sigma^{t-1},a^t,d^t],s^t} q_{k,s^t} [\sigma^{t-1},a^t,d^t,s^t]
\]

Then \( \min_{v \in V} \max_{u \in U} J_{I_{X,t}}(u, v) \) is equal to

\[
\min_{v \in V} \max_{w \in W} J_{I_{X,t}}(w, v).
\]
From the expression in (6.21), \( \tilde{J}_{I_{X,t}}(w, v) \) is a bilinear function of \( w, v : w'Av \) where \( A \) is a matrix with dimension \(|w| \times |v|\) and the elements are the coefficients of the corresponding product of element in \( w \) and element in \( v \). As discussed before, the constraint in \( W \) is denoted as \( Lw = l \). Let \( n \) be the dual variable to \( Lw = l \), then the dual problem of \( \max_{w \in W} w'Av \) is \( \min_{L'n \geq Av} l'n \). Since \( \tilde{J}_{I_{X,t}}(w, v) \) are bounded for all \( w, v \), by strong duality, replacing \( \max_{w \in W} w'Av \) with \( \min_{L'n \geq Av} l'n \), we can convert (6.22) to a LP problem

\[
\min_{v, n} \quad l'n \\
\text{s.t.} \quad \sum_{a'} v_{[\sigma^{t-1}, a'], d'} = 1, \forall [\sigma^{t-1}, a'] \succeq I_{X,t}; v \geq 0, \\
L'n \geq Av.
\]

Therefore, solving (6.23) can get the saddle point solution \( v \) in (6.22).

Notice \( \tilde{J}_{I_{X,t}}(w, v) \) is a bilinear function on \( w, v \) with probability conservation constraints. By theorem 6.1.6, it's equal to

\[
\max_{w \in W} \min_{u \in V} \tilde{J}_{I_{X,t}}(w, v).
\]

Denote the common optimal value of (6.22) and (6.24) as \( V_{I_{X,t}}(\beta^{I_{X,t}}) \). By substituting \( u_{I_{Y,t}, d'} \beta^{I_{X,t}-1}_{I_{Y,t}} \) in \( \tilde{J}_{I_{X,t}}(w, v) \), the resulting objective function has the form of \( J(x, y) \) in (6.5), therefore theorem 6.1.8 applied, \( V_{I_{X,t}}(\beta^{I_{X,t}}) \) can be represented by a finite set of support vectors. Furthermore, by theorem 6.1.6, the optimal solutions of outer optimization problem in (6.22) and (6.24), denoted as \( v^*, w^* \) are saddle point solutions. Because (6.22) are converted from (6.20) without changing \( v, v^* \) is also the saddle solution of (6.20).

Now we show how to solve (6.24) to get the support vector for a given \( \beta^{I_{X,t}} \).
Expand the objective function in (6.24), we have

$$\max_{w \in W} \min_{v \in V} \sum_{\sigma^{t-1} \Downarrow I^{X,t}} \beta^{I^{X,t}}_{\sigma^{t-1}} \sum_{a^t} \sum_{dT} v[\sigma^{t-1}, a^t, d^t] dt \sum_{k=1}^{K^{I^{X,t}, t+1}} \sum_{s^t} r[\sigma^{t-1}, a^t, d^t, s^t] q_k[\sigma^{t-1}, a^t, d^t, s^t]$$

$$= \max_{w \in W} \sum_{\sigma^{t-1} \Downarrow I^{X,t}} \beta^{I^{X,t}}_{\sigma^{t-1}} \sum_{a^t} \sum_{d^t} \sum_{k=1}^{K^{I^{X,t}, t+1}} \sum_{s^t} w_k^{I^{X,t+1}} r[\sigma^{t-1}, a^t, d^t, s^t] q_k[\sigma^{t-1}, a^t, d^t, s^t]$$

The equation is true because player \( Y \) minimizes the payoff by controlling \( v[\sigma^{t-1}, a^t, d^t] \). Define

$$m_{\sigma^{t-1}, a^t} := \min_{d^t} \sum_{k=1}^{K^{I^{X,t}, t+1}} w_k^{I^{X,t+1}} \sum_{s^t} r[\sigma^{t-1}, a^t, d^t, s^t] q_k[\sigma^{t-1}, a^t, d^t, s^t], \forall \sigma^{t-1}, a^t \not\in I^{X,t},$$

and use them as surrogate variables. Then (6.24) is equal to the following problem

$$\max_{w \in W, m} \sum_{\sigma^{t-1} \Downarrow I^{X,t}} \beta^{I^{X,t}}_{\sigma^{t-1}} \sum_{a^t} \sum_{d^t} m_{\sigma^{t-1}, a^t} \sum_{k=1}^{K^{I^{X,t}, t+1}} \sum_{s^t} r[\sigma^{t-1}, a^t, d^t, s^t] q_k[\sigma^{t-1}, a^t, d^t, s^t] \geq m_{\sigma^{t-1}, a^t}, \forall a^t, d^t, \sigma^{t-1}. \tag{6.25}$$

Let the optimal solution be \((w^*, m^*)\), define

$$q^* := \{q_{\sigma^{t-1}} | q_{\sigma^{t-1}} = \sum_{a^t} m_{\sigma^{t-1}, a^t} \}. \tag{6.26}$$

Then \( q^* \) is the support vector for the given \( \beta^{I^{X,t}} \), because the payoff function \( V_{I^{X,t}}(\beta) \) is equal to the optimal value of (6.25), which is

$$V_{I^{X,t}}(\beta) = \sum_{\sigma^{t-1}} \beta_{\sigma^{t-1}} \sum_{a^t} m_{\sigma^{t-1}, a^t} = \sum_{\sigma^{t-1}} \beta_{\sigma^{t-1}} q_{\sigma^{t-1}}^*.$$ 

One can solve (6.25) and obtain the support vector by (6.26). With \( w^* \), one can also obtain the equilibrium behavior strategy \( u^* \) on \( I^{X,t} \), as shown in the following theorem.
Theorem 6.2.4 Let \( w^* \) be an (part of) optimal solution to (6.25) and define \( \tilde{u}^* \) such that
\[
\tilde{u}^*_{I^x,t,a} = \sum_{k=1}^{K_{I^x,t+1}} u^*_k I^x,t+1, \quad \forall I^x,t+1 \succ I^x,t
\] (6.27)
then \( \tilde{u}^* \) is the equilibrium behavior strategy on \( I^x,t \) in the subgame \( G(I^x,t) \).

Proof: By the definition of \( \tilde{u}^* \), for any \( v \in V \), \( J_{I^x,t}(\tilde{u}^*, v) = \tilde{J}_{I^x,t}(w^*, v) \). Let \((v^*, u^*)\) be the saddle point solutions in (6.20), i.e,
\[
J_{I^x,t}(u^*, v^*) = \max_{u \in U} J_{I^x,t}(u, v^*) = \min_{v \in V} J_{I^x,t}(u^*, v).
\]
Because \( w^* \) is the saddle point solution in (6.24) and (6.25) is converted from (6.24), which has the same optimal value as (6.20). Then we have
\[
J_{I^x,t}(u^*, v^*) = \min_{v \in V} \tilde{J}_{I^x,t}(w^*, v) = \min_{v \in V} J_{I^x,t}(\tilde{u}^*, v).
\]
Because \( \tilde{J}_{I^x,t}(w, v) \) are converted from \( J_{I^x,t}(u, v) \) without changing \( v \), then \( v^* \) is also the saddle solution in (6.22), then
\[
J_{I^x,t}(\tilde{u}^*, v^*) = \tilde{J}_{I^x,t}(w^*, v^*) = J_{I^x,t}(u^*, v^*) = \max_{u \in U} J_{I^x,t}(u, v^*).
\]
As a result, we have
\[
J_{I^x,t}(\tilde{u}^*, v^*) = \max_{u \in U} J_{I^x,t}(u, v^*) = \min_{v \in V} J_{I^x,t}(\tilde{u}^*, v),
\]
which completes the proof.

The procedure to identify the sets of support vectors for subgame \( G(I^x,t) \) for all \( I^x,t, t = 1, \ldots, T \) is a one-pass backward method, similar to dynamic programming. We first evaluate the sets of support vectors \( Q_{I^x,t} \) for all the lowest level subgames \( G(I^x,T, \cdot) \). With \( Q_{I^x,T} \), one can obtain a support vector of subgame \( G(I^x,T-1, \beta^{T-1}) \) by solving (6.25). We will use Procedure 10 to obtain the set of support vectors. By
repeating this process until subgame $G(I^{X,t}, \cdot)$, one can identify the set of support vector $Q$ for $V(\beta^0)$. Procedure 9 summaries the above process. To evaluate the sets of support vectors $Q_{I^{X,t}}$ for all subgame $G(I^{X,t}, \cdot)$.

**Procedure 9** Identify the sets of support vectors for all subgame $G(I^{X,t}, \cdot)$.

Find the sets of support vectors $Q_{I^{X,t}}$ for subgame $G(I^{X,t}, \cdot)$ for all $I^{X,t}$.

for $t = T-1, \ldots , 1$ do
  for all subgame $G(I^{X,t}, \cdot)$ do
    With given $Q_{I^{X,t+1}}$ for any $I^{X,t+1} \succ I^{X,t}$, find the sets of support vectors $Q_{I^{X,t}}$ (Procedure 10)
  end for
end for

of support vectors $Q_{I^{X,t}}$ with given $Q_{I^{X,t+1}}$ for any $I^{X,t+1} \succ I^{X,t}$, we borrow the idea of Point-based Value Iteration (Section 2.5), as shown in Fig. 6.5.

We take the advantage of the nested information structure, which is similar to the structure of Partial Observed Markov Dynamic Process (POMDP) where player $X$'s strategy is equivalent to the control and player $Y$'s strategy is optimized based on player $X$'s strategy. The framework of this subroutine is shown in Procedure 10. Here the number of samples $M$ is proportional to the dimension of $\beta^0$.

Please note that in Procedure 10, for each $I^{X,t}$ we also maintain the set of weights
Procedure 10 Estimating the set of support vectors \( Q_{I^X,t} \)

- Initialize \( Q_{I^X,t} = \emptyset, W_{I^X,t} = \emptyset \).
- Sample \( M \) points of \( \beta_{I^X,t} \) independently.

\[
\text{for each } \beta_{I^X,t} \text{ do}
\]
- Solve (6.25), denote the optimal value as \( V_{I^X,t}(\beta_{I^X,t}) \).
  \[ \text{if } V_{I^X,t}(\beta_{I^X,t}) < \min_{q \in Q_{I^X,t}} \beta_{I^X,t} \cdot q \text{ then} \]
  - Denote the optimal solution of (6.25) as \( (w^*, m^*) \)
  - Construct its support vector \( q^* \) by (6.26) with \( m^* \).
  - add \( q^* \) to \( Q_{I^X,t} \), \( w^* \) to \( W_{I^X,t} \).
\[
\text{end if}
\]
\[
\text{end for}
\]

of lower level support vectors \( W_{I^X,t} \), which will be used to retrieve the equilibrium strategies for the original game in the next phase.

6.2.3 Retrieving the Global Saddle Point Strategies

After identifying the sets of support vectors for all subgames \( G(I^X,t, \cdot) \), we will retrieve the equilibrium strategies for the original game for a given initial distribution \( \beta^0 \). This process starts from the top level subgame \( G(I^X,1, \beta^0) \). It obtains the first interaction saddle point strategies \( u_{I^X,1,a}^* \) and \( v_{I^Y,1,d}^* \) by solving (6.25) (convert \( w^* \) to \( u^* \) via (6.27)) and (6.22). Then the realization probability of any history \( \sigma^1 = [s^0, a^1, d^1, s^1] \) in any \( I^{X,2} \) can be calculated by

\[
\beta_{I^{X,2}}^{[s^0, a^1, d^1, s^1]} = \beta_{I^{X,1}}^{s_0} u_{I^X,1,a}^* v_{I^Y,1,d}^* r_{[s^0, a^1, d^1, s^1]},
\]

which will be normalized within each \( I^{X,2} \) for \( P_{I^X,2} > 0 \) and then used as the initial distribution in \( G(I^{X,2}, \cdot) \). For \( t \geq 2 \), the behavior strategies on \( I^{Y,t} \) can be obtained by solving the corresponding (6.23) for that subgame \( G(I^{X,t}, \beta_{I^{X,t}}) \). However, the behavior strategy on \( I^{X,t} \) is calculated from \( u_{I^{X,t-1}}^* \) and \( W_{I^{X,t}} \). Once the behavior strategy for \( I^{X,t} \) is ready, the realization probability of \( \beta_{I^{X,t+1}} \) for any \( I^{X,t+1} \rightarrow I^{X,t} \) is

\[
\beta_{I^{X,t+1}}^{[\sigma^{t-1}, a^t, d^t, \sigma^t]} = \beta_{I^{X,t}}^{[\sigma^{t-1}, a^t, d^t, \sigma^t]} u_{I^{X,t-1}, a^t}^* v_{I^{Y,t-1}, a^t}^* r_{[\sigma^{t-1}, a^t, d^t, \sigma^t]}.
\]
We can also calculate the realization probability for an information set

\[ P_{I_{x,t+1}} = \sum_{\sigma^{t-1}, \sigma^t} \beta_{\sigma^{t-1}, \sigma^t}^{I_{x,t}} u_{I_{x,t}, \sigma^t} v_{[\sigma^{t-1}, \sigma^t], dt} r_{[\sigma^{t-1}, \sigma^t], dt, \sigma^t}. \]  

(6.29)

For any \( I_{x,t+1} = [I_{x,t}, a^t, d^t] \) with \( P_{I_{x,t+1}} > 0 \), we further normalize the conditional distributions.

\[ \tilde{\beta}^{I_{x,t+1}}_{[\sigma^{t-1}, a^t, d^t, \sigma^t]} = \frac{\beta^{I_{x,t+1}}_{[\sigma^{t-1}, a^t, d^t, \sigma^t]}}{\sum_{\tilde{\sigma}^{t-1}, \tilde{\sigma}^t} \beta^{I_{x,t}}_{[\tilde{\sigma}^{t-1}, a^t, d^t, \tilde{\sigma}^t]}} \]

\[ = \frac{\beta^{I_{x,t}}_{\sigma^{t-1}, a^t, d^t}}{\sum_{\tilde{\sigma}^{t-1}, \tilde{\sigma}^t} \beta^{I_{x,t}}_{\tilde{\sigma}^{t-1}, a^t, d^t}}. \]  

(6.30)

Note in this step, we ignore the subgames with zero realization probability. This process continues until it obtains the behavior strategies on all on-the-path information sets (information sets with positive realization probability given the strategies on upper level information sets). The idea of the top down retrieval process is shown in Fig. 6-6.

We obtain \( \omega^{*I_{x,t}} \) by solving (6.25) for the first one stage subgame, which is equivalent to the saddle point strategy by theorem 6.2.4. Now we show how to calculate...
from \( w^{*I_{X,t-1}} \) in a top down process. Notice \( w^{*I_{X,t-1}} \) is a set of weights on the support vectors in \( Q_{I_{X,t}} \) for all \( I_{X,t} \succ I_{X,t-1} \).

For any \( I_{X,t} = [I_{X,t-1}, a^{t-1}, d^{t-1}] \), denote its support vector set as

\[
Q_{I_{X,t}} = \{ q_{I_{X,t},k} \},
\]

and the corresponding set of weights as

\[
W_{I_{X,t}} = \{ w_{I_{X,t},k} \},
\]

where each \( w_{I_{X,t},k} \) corresponding to a \( q_{I_{X,t},k} \), which is obtained in Procedure (10) while solving the problem of (6.25) for subgame \( G([I_{X,t-1}, a^{t-1}, d^{t-1}], \cdot) \).

Denote the weight corresponding to the support vector \( q_{I_{X,t},k} \) in \( w^{*I_{X,t-1}} \) as \( w_{k_{I_{X,t}}} \) where \( I_{X,t} \) depends on \( a^{t-1}, d^{t-1} \). By theorem 6.2.4, we can calculate the equilibrium behavior strategy \( u_{I_{X,t-1}}^{*} \) on \( I_{X,t-1} \) from \( w^{*I_{X,t-1}} \),

\[
u_{I_{X,t-1},a^{t-1}}^{*} = \sum_{k=1}^{K_{I_{X,t}}} w_{k_{I_{X,t}}}^{*I_{X,t-1}}.
\]

For any \( I_{X,t} = [I_{X,t-1}, a^{t-1}, d^{t-1}] \) such that \( u_{I_{X,t-1},a^{t-1}}^{*} > 0 \), we can normalize the weights within \( I_{X,t} \) as

\[
\lambda_{k_{I_{X,t}}}^{I_{X,t}} := w_{k_{I_{X,t}}}^{*I_{X,t-1}} / u_{I_{X,t-1},a^{t-1}}^{*}, \forall k = 1, \ldots, K_{I_{X,t}}.
\]

Then \( w^{*I_{X,t}} \) is calculated as

\[
\sum_{k=1}^{K_{I_{X,t}}} \lambda_{k_{I_{X,t}}}^{I_{X,t}} w_{I_{X,t},k}.
\]

The following discussion is to justify that \( w^{*I_{X,t}} \) calculated in (6.35) is the (part of) optimal solution of (6.25) corresponding to subgame \( G(I_{X,t}, \beta^{I_{X,t}}) \), thus it's equivalent to a saddle point strategy.
In problem (6.25), define \( q(w) := \{ q(w)_{\sigma t-1} | \sigma t-1 \supset I X,t \} \), where \( q(w)_{\sigma t-1} \) is

\[
q(w)_{\sigma t-1} := \sum_{a_t} \min_{v} \sum_{d_t} v_{[\sigma t-1, a_t], d_t} \sum_{k=1}^{K I X,t+1} w_{k}^{I X,t+1} \sum_{s_t} r_{[\sigma t-1, a_t, d_t, s_t]} q_{k, [\sigma t-1, a_t, d_t, s_t]}.
\]

By this definition, \( q(w)_{\sigma t-1} \) is the expected payoff on node \( \sigma t-1 \) when player \( Y \) optimizes its strategy for a given strategy \( w \) of player \( X \).

**Lemma 6.2.5** For any \( \sigma t-1 \supset I X,t \), \( q(w)_{\sigma t-1} \) is a concave function on \( w \).

**Proof:** For each \( a_t \), we are minimizing over a finite set of linear functions of \( w \), which is a concave function. The sum of concave functions is concave. \( \blacksquare \)

**Lemma 6.2.6** Let \( w^{I X,t,k} \) be a solution in (6.32) and \( q^{I X,t,k} \) is the corresponding support vector in (6.31), then

\( q^{I X,t,k} = q(w^{I X,t,k}) \).

**Proof:** Since each \( w^{I X,t,k} \) is the saddle point strategy when solving the problem (6.25) for subgame \( G(I X,t, \beta_t) \), and \( q^{I X,t,k} \) is calculated according to (6.26), which is the expected payoff when player \( Y \)'s optimizes its strategy for the given \( w^{I X,t,k} \), therefore the conclusion is true. \( \blacksquare \)

For any \( I X,t \supset [I X,t-1, a t-1, d t-1] \), with \( \lambda^* \) calculated in (6.34), define

\[
\bar{q}_{I X,t} := \sum_{k=1}^{K I X,t} \lambda_{k}^{*} q^{I X,t,k}.
\]

**Lemma 6.2.7** Let \( w^{* I X,t-1} \) be the solution of problem (6.25) corresponding to the subgame \( G(I X,t-1, \beta^{I X,t-1}) \). For a next level subgame \( G(I X,t, \beta^{I X,t}) \) with the initial distribution \( \beta^{I X,t} \) obtained according to (6.30), \( \bar{q}_{I X,t} \), defined in (6.37) is a support vector in this subgame.
Proof: Assume the lemma is not true, then there is another convex combination of support vectors in the subgame \( G(I^{X,t}, \beta^{I^{X,t}}) \), \( \tilde{q} := \sum_{k=1}^{K_I^{X,t}} \tilde{\lambda}_k^{I^{X,t}} q^{I^{X,t},k} \), such that

\[
\beta^{I^{X,t}} \cdot \tilde{q} > \beta^{I^{X,t}} \cdot q^{I^{X,t}}.
\]

When player \( X \) replaces \( \lambda_k^{I^{X,t}} \) with \( \tilde{\lambda}_k^{I^{X,t}} \) and keeps other weights the same, its payoff in this subgame is

\[
\beta^{I^{X,t}} \cdot \tilde{q},
\]

which is better than the payoff when it chooses \( w_*^{I^{X,t-1}} \)

\[
\beta^{I^{X,t}} \cdot q^{I^{X,t}}.
\]

This contradicts with \( w_*^{I^{X,t-1}} \) is the saddle point strategy in \( G(I^{X,t-1}, \beta^{I^{X,t-1}}) \). Therefore the assumption is false and \( q^{I^{X,t}} \) is a support vector in the subgame \( G(I^{X,t}, \beta^{I^{X,t}}) \).

\[ \blacksquare \]

**Theorem 6.2.8** Let \( w_*^{I^{X,t}} \) be calculated in (6.35) and \( q(I^{X,t},q(w))_{\sigma} \) be defined as in (6.37) and (6.36), respectively, then

\[ P1 \] For any \( \sigma \geq I^{X,t} \), \( q_{\sigma-1}(w_*^{I^{X,t}}) \geq q_{I^{X,t},\sigma-1} \); and \( q_{\sigma-1}(w_*^{I^{X,t}}) = q_{I^{X,t},\sigma-1} \) if \( \beta_{\sigma-1}^{I^{X,t}} > 0 \).

\[ P2 \] \( w_*^{I^{X,t}} \) is a saddle point strategy in subgame \( G(I^{X,t}, \beta^{I^{X,t}}) \).

Proof: By lemma 6.2.6, we have

\[
q_{I^{X,t}} = \sum_{k=1}^{K_I^{X,t}} \lambda_k^{I^{X,t}} q^{I^{X,t},k} = \sum_{k=1}^{K_I^{X,t}} \lambda_k^{I^{X,t}} q(w_*^{I^{X,t},k}).
\]

By lemma 6.2.5, \( q_{\sigma-1}(w) \) is concave on \( w \), therefore,

\[
q_{I^{X,t},\sigma-1} \leq q_{\sigma-1}(\sum_{k=1}^{K_I^{X,t}} \lambda_k^{I^{X,t}} w_*^{I^{X,t},k}) = q_{\sigma-1}(w_*^{I^{X,t}}).
\]

\[ (6.38) \]
On the other side, by lemma 6.2.7, \( \bar{q}_{I_1,t} \) is a support vector for subgame \( G(I_1^X,t, \beta I_1^X,t) \), then
\[
\beta I_1^X \cdot \bar{q}_{I_1,t} = V_{I_1,t}(\beta I_1^X,t) = \beta I_1^X \cdot q(w^* I_1^X,t).
\]
\( V_{I_1,t}(\beta I_1^X,t) = \beta I_1^X \cdot q(w^* I_1^X,t) \) means given \( w^* I_1^X,t \), player \( Y \) can not get better payoff by changing its strategy, therefore \( w^* I_1^X,t \) is the saddle point strategy in \( G(I_1^X,t, \beta I_1^X,t) \).

Furthermore, consider \( \beta I_1^X \cdot q(w^* I_1^X,t) = \beta I_1^X \cdot \bar{q}_{I_1,t} \) and (6.38), notice \( \beta I_1^X \) is non-negative, then
\[
q_{\sigma = t-1}(w^* I_1^X,t) = \bar{q}_{I_1,t, \sigma = t-1}, \quad \forall \beta I_1^X > 0.
\]

The top down retrieval process is summarized in Procedure (11).

**Procedure 11** Retrieve the global saddle-point strategies \( \bar{u}, \bar{v} \) given \( \beta^0 \).

- Solve (6.23), denote the optimal solution as \( \bar{u}_{\beta}, \bar{v} \).
- Solve (6.25), denote the (part of) optimal solution \( w \) as \( w^* I_1^X,t \).
- Retrieve \( \bar{u}_{I_1^X,t, a^1} \) with \( w^* I_1^X,t \) via (6.27).
- Calculate the distribution \( \beta I_1^X \) by (6.28).
- for any \( t = 2, \ldots, T \) do
  - for any \( I_1^X,t \) with \( P_{I_1^X,t} > 0 \) do
    - Solve (6.23) with \( \beta I_1^X \) from the upper level subgame, denote the optimal solution as \( \bar{u}_{\beta}, \bar{v} \), \( \forall d' \).
    - Calculate \( w^* I_1^X,t \) from \( w^* I_1^X,t-1 \) and \( W_{I_1^X,t} \) via (6.35).
    - Retrieve \( \bar{u}_{I_1^X,t} \) with \( w^* I_1^X,t \) in (6.33).
    - Calculate the realization probability \( P_{I_1^X,t+1} \) by (6.29).
      - for any \( I_1^X,t+1 \) with \( P_{I_1^X,t+1} > 0 \) do
        - Calculate the initial distribution \( \beta I_1^X,t+1 \) for \( G(I_1^X,t+1, \beta I_1^X,t+1) \) by (6.30).
      - end for
    - end for
  - end for
- end for
- Output players’ behavior strategies \( \bar{u}, \bar{v} \).

By theorem (6.2.3), the behavior strategies obtained in Procedure (11) for the first interaction of each subgame are part of the saddle point strategies of the original game on that interaction. Note that combining these local saddle point strategies
together does not necessarily yield the global saddle strategies, i.e., if we obtain $w^{*I^{X,t}}$ by solving one stage subgames $G(I^{X,t}, \beta^{I^{X,t}})$ alone and put $w^{*I^{X,t}}$ for all $I^{X,t}$ together, the resulting strategy is not necessarily a saddle point strategy in the multi-stage game, as shown with an example in the subsection 6.2.4. Therefore we need the following theorem to guarantee the outputs of Procedure (11) are the global equilibrium.

**Theorem 6.2.9** The strategies $\tilde{u}, \tilde{v}$ obtained in Procedure (11) are the saddle-point strategies of the original game.

**Proof:** See Appendix.

6.2.4 Subgame Decomposition on a Two-interaction Game

We apply the subgame decomposition method for a two-interaction extensive form game to illustrate the process of this method. In order to make it easy to track, we construct the payoffs and the structure of the game as simple as possible, as shown in Fig. 6·7. This is a zero-sum game between player $X$ and player $Y$, whose strategies are denoted as $x^t, y^t$ at time $t = 1, 2$. The game starts at Nature selecting the underlying state under the initial distribution $\beta^0 = (1/2, 1/2)$ at the top level. Then player $X$ and player $Y$ act by turn with player $X$ moves first. The actions of players are represented by lines with solid lines (after Nature) for player $X$ and dash ones for player $Y$. Payoffs to player $X$ are shown at the bottom. Two plays associated with payoffs $-10$ after player $X$'s first action are constructed to provide an alternate choice for player $X$. Player $X$ does not know the exact underlying state, therefore its information sets are represented by ellipses, which contains the sequences that player $X$ can not tell the difference.

The subgame decomposition method first decomposes the game. We get the first level subgame $G^1$ and two lower level subgames $G^{21}, G^{22}$, whose ranges are represented by dash rectangles in Fig. (6·7). Then in Procedure (9), the method evaluates the support vectors $Q_{21}, Q_{22}$ for subgames $G^{21}, G^{22}$ via Point-based value iteration of
Figure 6-7: Subgame decomposition on an extensive form game.

Procedure (10). In subgame $G^{21}$, because player $Y$ is a minimizer, it will definitely not choose 10 at the second stage. It's easy to find the support vectors are

$$q_1^{21} = (1, 0), q_2^{21} = (0, 2).$$

The corresponding weights for those support vectors are

$$w_1^{21} = (0, 1, 0, 0), w_2^{21} = (0, 0, 0, 1).$$

Each $w_k^{21}$ represents the weights of $G^{21}$'s lower level support vectors

$$(10, 10), (1, 0), (10, 10), (0, 2),$$

which are the payoffs since $G^{21}$ is at the bottom. Similarly, the support vectors for $G^{22}$ are

$$q_1^{22} = (2, 0), q_2^{22} = (0, 1).$$

and the corresponding weights are

$$w_1^{22} = (0, 1, 0, 0), w_2^{22} = (0, 0, 0, 1).$$
Since $G^1$ is the highest level subgame with a given initial distribution, we don't need to estimate its value function. The method then goes to the strategy retrieval phase. It solves the problems of (6.22) and (6.25) with its lower level support vectors as $q_i^{2k}$ where $k, i = 1, 2$ and initial distribution as $\beta^0$. The saddle point payoff for this game is $v^* = 2/3$ and the saddle point strategies for player $Y$ at the first stage are

$$v^{1*} = (2/3, 1/3; 1/3, 2/3)$$

where each element in $v^{1*}$ represents the probability of taking an action. By solving (6.25), the optimal weights of the 5 support vectors are

$$w^{1*} = (2/3, 1/3; 1/3, 2/3; 0),$$

where the first two elements correspond to support vectors $q_k^{21}, k = 1, 2$ and the next two correspond to $q_k^{22}, k = 1, 2$. The last one was for the payoff $(-10, -10)$. Based on $w^{1*}$, one can derive $u^{*1} = (1, 0), \lambda^{*21} = (2/3, 1/3)$ and $\lambda^{*22} = (1/3, 2/3)$. With $\beta^0, v^{1*}, u^{1*}$, we have the realization probabilities for nodes in the initial information sets of lower level subgame via (6.29). Then the initial distributions for $G^{21}$ and for $G^{22}$ are

$$\beta^{21} = (2/3, 1/3), \beta^{22} = (1/3, 2/3),$$

respectively. Based on $w^{1*}$, one can derive the weights of support vectors of subgame $G^{21}$ via (6.34): $\lambda^{*21} = (2/3, 1/3)$. Using $\lambda^{*21}$ to weight the solutions in $W^{21}$ via (6.35), we have $w^{21*} = \lambda_1^{*21} * w_1^{21} + \lambda_2^{*21} * w_2^{21} = (0, 2/3; 0, 1/3)$. From $w^{21*}$ we have $u^{*21} = (2/3, 1/3)$, meaning player $X$ chooses left with probability 2/3. Notice player $Y$ will definitely not choose 10, the expected payoffs $\overline{q}_{21}$ for nodes in the initial information set of subgame $G^{21}$ is $(2/3, 2/3)$. Similarly, we get

$$u^{*22} = (1/3, 2/3)$$
the expected payoff values for nodes in $G^{22}$ are $\tilde{q}_{22} = (2/3, 2/3)$. One can verify that with these strategies, the game achieves equilibrium.

It's interesting to note that if we obtained the local saddle point strategy $u^{I^{X,t}}$ within subgame $G(I^{X,t}, \beta')$ by solving (6.25) alone, instead of calculating $w^{I^{X,t}}$ via (6.34) and (6.35) as in Procedure (11), the game may fail to achieve global equilibrium. For instance, in the subgame $G^{21}$, if we solve (6.25), one possible strategy is $\tilde{w}^{21} = (0, 1, 0, 0)$. Notice player $Y$ will never choose the actions leading to payoff 10, the expected payoff under $\tilde{w}^{21}$ is 2/3, the same as the expected payoff under $w^{*21}$. Therefore $\tilde{w}^{21}$ is a local saddle point strategy. Changing $w^{21*}$ to $\tilde{w}^{21}$, the expected payoffs for nodes within the subgame are $\tilde{q}_{21} = (1, 0)$. Compared with the expected payoffs of $\tilde{q}_{22} = (2/3, 2/3)$ for nodes within another subgame $G^{22}$, player $Y$ can take advantage over that by changing its first stage strategy as $\tilde{v}^1 = (0, 1, 1, 0)$, then the new payoff is 1/3, less than the saddle point value of 2/3, therefore the game can not achieve equilibrium by these strategies.

### 6.3 Other Nested Information Games

In the previous nested information game, the sequence of players within each interaction is

attacker $\rightarrow$ defender $\rightarrow$ nature

This is not a necessary requirement for our subgame decomposition method. As long as the zero-sum game is nested information, our method can decompose it with subgames starting with the information sets of the less-informed player, solves these subgames and retrieves the saddle point strategy of the original game.

Here we propose several interesting extensions for other dynamic games of network interdiction problems. Notice the foundation of our method is the subgame decomposition, allowing the game to be decomposed into subgames with one-stage
less. One can implement the backward support vectors estimation and the forward strategies reconstruction in the same approach as for the basic model. Therefore in each extension, we will only focus on describing the subgame decomposition.

**Defender observes the updated network state before its move:** In some variations of the network interdiction game, the attacker interdicts the network, which has random outcomes controlled by Nature. Assume that the defender acts after observing the new network state, then the sequence of actions is

attacker → nature → defender.

Adapting the same notations from the basic model, the history of the game, the information sets corresponding to the attacker, the defender and Nature at time \( t \) are

\[
\sigma^t = [s^0, a^1, s^1, d^1, \ldots, a^t, s^t, d^t], t = 1, \ldots, T - 1;
\]
\[
I_{X,t} = [a^1, d^1, \ldots, a^{t-1}, d^{t-1}], t = 1, \ldots, T;
\]
\[
I_{Z,t} = [s^0, a^1, s^1, d^1, \ldots, a^{t-1}, a^t], t = 1, \ldots, T.
\]
\[
I_{Y,t} = [s^0, a^1, s^1, d^1, \ldots, a^{t-1}, s^t], t = 1, \ldots, T - 1.
\]

Since the game does not end with the final history \( \sigma^{T-1} \), denote the play of the game as

\[
\delta^T = [s^0, a^1, s^1, d^1, \ldots, a^{T-1}, s^{T-1}, d^{T-1}, a^T, s^T].
\]

Let \( u, v, r \) denotes the behavior strategies for the attacker, the defender and Nature as in the basic model, and \( \beta^0 \) be the initial probability on \( s^0 \), then the realization probability for a play \( \delta^T = [s^0, \bar{a}^1, \bar{s}^1, \bar{d}^1, \ldots, \bar{a}^{T-1}, \bar{s}^{T-1}, \bar{d}^{T-1}, \bar{a}^T, \bar{s}^T] \) is

\[
\text{Prob}\{\delta^T|u,v,r,\beta^0\} := \beta^0 u_{jx,1,\bar{a}^1} r_{jz,1,\bar{s}^1} v_{jy,1,\bar{d}^1} \prod_{t=2}^{T-1} u_{jx,t,\bar{a}^t} r_{jz,t,\bar{s}^t} v_{jy,t,\bar{d}^t} u_{jx,T,\bar{a}^T} r_{jz,T,\bar{s}^T}. 
\]
Define the attacker's (player $X$'s) and the defender's (player $Y$'s) realization plan as

$$x := \{x_{I^{X,t},a^t} | \forall t = 1, \ldots, T, \forall I^{X,t}, a^t\},$$

$$y := \{y_{I^{Y,t},d^t} | \forall t = 1, \ldots, T - 1, \forall I^{Y,t}, d^t\},$$

where $x_{I^{X,t},a^t}, y_{I^{Y,t},d^t}$ are defined recursively by

$$x_{I^{X,t},a^t} = u_{I^{X,t},a^t}, \quad \forall a^t;$$

$$x_{I^{X,t},a^t} = x_{I^{X,t-1}(I^{X,t}),a^t-1(I^{X,t})} u_{I^{X,t},a^t}, \quad \forall t > 1, I^{X,t}, a^t;$$

$$y_{I^{Y,t},d^t} = \beta^0_{s(I^{Y,t})} v_{I^{Y,t},d^t}, \quad \forall I^{Y,t}, d^t;$$

$$y_{I^{Y,t},d^t} = y_{I^{Y,t-1},d^t-1} v_{I^{Y,t},d^t}, \quad \forall t > 1, I^{Y,t}, d^t.$$

Then the constraints on $x, y$ are

$$\sum_{a^1} x_{I^{X,t},a^t} = 1; \quad \sum_{a^1} x_{I^{X,t},a^t} = x_{I^{X,t-1}(I^{X,t}),a^t-1(I^{X,t})}, \forall t > 1, I^{X,t}; \quad (6.39)$$

$$\sum_{d^t} y_{I^{Y,t},d^t} = \beta^0_{s(I^{Y,t})}, \forall I^{Y,t}; \quad \sum_{d^t} y_{I^{Y,t},d^t} = y_{I^{Y,t-1}(I^{Y,t}),d^t-1(I^{Y,t})}, \forall t > 1, I^{Y,t}. \quad (6.40)$$

Let $X$ be the feasible set of $x$ and $Y(\beta^0)$ be the feasible set of $y$, representing the above constraints on $x, y$, which written them concisely as

$$X := \{x \geq 0 | Ex = e\}; \quad Y(\beta^0) := \{y \geq 0 | Fy = B\beta^0\}. \quad (6.41)$$

where $E, F, e, B$ are matrices (or vectors) derived from the coefficients of the constraint equations, which are similar to the basic model.

Denote $r_{\delta^T} := \prod_{t=1}^{T} r_{I^{X,t}(\delta^T),a^t(\delta^T)}$, then the realization probability of a play $\delta^T$ is

$$\text{Prob}(\delta^T | x, y) := x_{I^{X,T}(\delta^T),a^T(\delta^T)} y_{I^{Y,T-1}(\delta^T),d^{T-1}(\delta^T)} r_{\delta^T}.$$
Therefore the expected payoff under strategies $x, y$ is

$$J(x, y) := \sum_{\delta^T} x_{1X,T}(\delta^T), a^T(\delta^T) y_{1Y,T-1}(\delta^T) d_{T-1}(\delta^T) r_{\delta^T} c_{\delta^T},$$

(6.42)

where $c_{\delta^T}$ is the payoff to the attacker when the game ends with $\delta^T$. Then the problems of finding equilibrium strategies can be formulated as

$$\max_{x \in X} \min_{y \in Y(\beta^0)} J(x, y), \quad \min_{y \in Y(\beta^0)} \max_{x \in X} J(x, y),$$

(6.43)

where $X, Y(\beta^0)$ are feasible sets of $x, y$ based on their definitions. Since $c_{\delta^T} r_{\delta^T}$ for all $\delta^T$ are known constants, then $J(x, y)$ is a bilinear function on $x, y$, the same formulations as in (6.7) and (6.6). Furthermore, the constraints of $x, y$ in (6.39) and (6.40) represents the same probability conservations as in (6.7) and (6.6). Then theorem 6.1.6 applies, we have that the optimal solutions in (6.43) are the saddle point strategies of the game. Let $V(\beta^0)$ be the saddle point value in (6.43), By theorem 6.1.8, there is a finite set of vectors $Q$, such that

$$V(\beta^0) = \max_{q \in Q} \beta^T q, \quad \forall \beta^0 \in [0, 1]|S|.$$

We’re going to decompose this game into subgames by separating the players’ first strategies from their remaining strategies. For any play $\delta^T$, the realization probability is

$$\beta^0 = \prod_{t=2}^{T-1} u_{I_{X,t} \ast a^t I_{Y,t}, d^t I_{Z,t}, s^t} (\prod_{t=2}^{T-1} u_{I_{X,t} \ast a^t I_{Y,t}, d^t I_{Z,t}, s^t}) u_{I_{X,T} \ast a^T I_{Z,T}, s^T},$$

where $I_{X,t}, a^t, I_{Y,t}, d^t, I_{Z,t}, s^t, t = 1, \ldots, T$ are all on the path of $\delta^T$. Define the first
interaction strategies before $I^{X,2}$ as
\[
x^1 := \{ x_{a1}^1 | x_{a1}^1 := u_{I^{X,1},a1} \forall a1 \},
\]
\[
y^1 := \{ y_{I^{Y,1},d1}^1 | y_{I^{Y,1},d1}^1 := \beta^0_{s0(I^{Y,1})} v_{I^{Y,1},d1}, \forall I^{Y,1}, d1 \}.
\]

And define the remaining strategies after $I^{X,2}$ as
\[
x^r := \{ x_{I^{X,t},a1}^r | \forall I^{X,t}, a1, t > 1 \}; \quad y^r := \{ y_{I^{Y,t},d1}^r | \forall I^{Y,t}, d1, t > 1 \}
\]
with
\[
x_{I^{X,t},a1}^r := u_{I^{X,t},a1} \prod_{\tau=2}^{t-1} u_{I^{X,\tau}(I^{X,t},a1),a1}(I^{X,t},a1), \forall I^{X,t}, a1;
\]
\[
y_{I^{Y,t},d1}^r := y_{I^{Y,1}(I^{Y,t}),d1}(I^{Y,t},d1) \prod_{\tau=2}^{t-1} y_{I^{Y,\tau}(I^{Y,t}),d1}(I^{Y,t},d1), \forall I^{Y,t}, t > 2.
\]

Based on their definition, the probability constraints on $x^1, y^1, x^r, y^r$ are
\[
\sum_{a1} x_{a1}^1 = 1; \quad \sum_{d1} y_{I^{Y,1},d1}^1 = \beta^0_{s0(I^{Y,1})}, \forall I^{Y,1};
\]
\[
\sum_{d2} y_{I^{Y,2},d2}^r = y_{I^{Y,1}(I^{Y,2}),d2}(I^{Y,2}), \forall I^{Y,2}; \quad \sum_{a2} x_{I^{X,2},a2}^r = 1, \forall I^{X,2};
\]
\[
\sum_{a1} x_{I^{X,t},a1}^r = x_{I^{X,t-1}(I^{X,t}),a1}(I^{X,t},a1), \forall I^{X,t}, t > 1;
\]
\[
\sum_{d1} y_{I^{Y,t},d1}^r = y_{I^{Y,t-1}(I^{Y,t}),d1}(I^{Y,t},d1), \forall I^{Y,t}, t > 2.
\]

Denote the feasible sets for $x^1, x^r$ as $X^1, X^r$ respectively. Notice the constraints of $y^r$ depend on $y^1$ and the constraints of $y^1$ depend on $\beta^0$, denote the feasible set of $y^r$ as $Y^r(y^1)$ and the feasible set of $y^1$ as $Y^1(\beta^0)$.

Denote $r_{\delta^T} := \prod_{\tau=2}^{T} r_{I^{Z,\tau}(\delta^T),s\tau(\delta^T)}$, then the realization probability of $\delta^T$ can be
written as
\[ \sum_{\delta T} x_{a1}^i(\delta T) T_{\delta T} x_{\delta T}^r x_{\delta T}^T a(\delta T) y_{\delta T}^T \delta T \delta T. \]

Given \( x^1, y^1, x^r, y^r \) and \( \beta^0 \), the expected payoff is
\[ \bar{J}(x^1, y^1, x^r, y^r) := \sum_{\delta T} x_{a1}^i(\delta T) T_{\delta T} x_{\delta T}^r x_{\delta T}^T a(\delta T) y_{\delta T}^T \delta T \delta T. \]

\[ = \sum_{I^X,2} x_{a1}^i(\delta T) x_{I^X,2}^r x_{I^X,2}^T a(\delta T) y_{I^X,2}^T \delta T \delta T, \]

where the second sum groups plays \( \delta T \) into the different information sets \( I^X,2 \) that they belong to. By the definition of \( x^1, x^r \), maximizing \( x \) in \( J(x, y) \) is equivalent to maximizing \( x^1, x^r \) in \( \bar{J}(x^1, y^1, x^r, y^r) \). Similarly by the definition of \( y^1, y^r \), minimizing \( y \) in \( J(x, y) \) is equivalent to minimizing \( y^1, y^r \) in \( \bar{J}(x^1, y^1, x^r, y^r) \), therefore (6.43) is equal to
\[ J_{MMmM} := \max_{x^1 \in X^1, x^r \in X^r} \min_{y^1 \in Y^1(\beta^0), y^r \in Y^r(y^1)} \bar{J}(x^1, y^1, x^r, y^r), \]
(6.44)
\[ J_{mndMM} := \min_{y^1 \in Y^1(\beta^0), y^r \in Y^r(y^1)} \max_{x^1 \in X^1, x^r \in X^r} \bar{J}(x^1, y^1, x^r, y^r). \]
(6.45)

Notice \( V(\beta^0) \) is defined as the optimal value of (6.43), then we have
\[ J_{MMmM} = J_{mndMM} = V(\beta^0). \]

Consider the following optimization problems
\[ J_{mMmM} := \min_{y^1 \in Y^1(\beta^0)} \max_{x^1 \in X^1, x^r \in X^r(y^1)} \min_{y^r \in Y^r(\beta^0)} \bar{J}(x^1, y^1, x^r, y^r); \]
(6.46)
\[ J_{MMmM} := \min_{x^1 \in X^1} \max_{y^1 \in Y^1(\beta^0), y^r \in Y^r(y^1)} \min_{x^r \in X^r} \bar{J}(x^1, y^1, x^r, y^r), \]
(6.47)

where (6.46) is the result of exchanging the optimization orders of \( x^1 \) and \( y^r \) in (6.45) and (6.47) is the result of exchanging the optimization orders of \( y^1 \) and \( x^r \) in (6.44). With the same deduction as that for the basic model, we can show that
Theorem 6.3.1 The saddle point strategies \( u^*, y^* \) in (6.47) and (6.46) are saddle point strategies of the first interaction in the original game.

Therefore we can apply the subgame decomposition method for this game as for the basic model.

**Attacker has noisy observation on flows:** In some applications, attacker may have noisy observations on the flows, i.e., the defender’s actions. This noisy observation depends on previous actions, or simply on whether there is flow on that arc. The attacker has random false observations with probabilities known by both the attacker and the defender. These observations can be taken as actions of Nature, denoted as \( \sigma^t \), for \( t = 1, \ldots, T \). Then the sequence of actions is

\[
\text{attacker} \rightarrow \text{new state} \rightarrow \text{defender} \rightarrow \text{observation}.
\]

Then the histories and the information sets in this game can be written as

\[
\sigma^t = [s^0, a^1, s^1, d^1, o^1, \ldots, a^t, s^t, d^t, o^t], t = 1, \ldots, T - 1;
\]
\[
I^X,t = [a^1, o^1, \ldots, a^{t-1}, o^{t-1}], t = 1, \ldots, T;
\]
\[
I^Y,t = [s^0, a^1, s^1, d^1, o^1, \ldots, a^t, s^t], t = 1, \ldots, T - 1;
\]
\[
I^Z,t = [s^0, a^1, s^1, d^1, o^1, \ldots, a^t, s^t], t = 1, \ldots, T - 1;
\]
\[
I^O,t = [s^0, a^1, s^1, d^1, o^1, \ldots, a^t, s^t, d^t], t = 1, \ldots, T - 1;
\]

where \( I^O,t \) is the information set of Nature at which it controls the attacker’s observations. Since the game does not end with the final history \( \sigma^{T-1} \), denote the play of the game as

\[
\delta^T = [s^0, a^1, s^1, d^1, o^1, \ldots, a^{T-1}, s^T, d^T].
\]

Let \( u, v, r, n \) denotes the behavior strategies for the attacker, the defender, the state transition and the noisy observation respectively and \( \beta^0_n \) be the initial probabil-
ity on $s^0$, then given a play $\delta^T = [s^0, \delta^1, s^1, \delta^1, \ldots, s^T, \delta^{T-1}, \delta^{T-1}, \delta^T, \delta^T]$, its realization probability is

$$Prob\{\delta^T|u,v,r,\beta^0\} := \beta^0_0 u_{I,1,1} r_{I,1,1,v} u_{I,1,1} r_{I,1,1,v}$$

Define the attacker's (player X's) and the defender's (player Y's) realization plan as

$$x := \{x_{I,x,t,a}^1|\forall t = 1, \ldots, T, \forall I^{X,t}, a^t\},$$

$$y := \{y_{I,y,t,d}^1|\forall t = 1, \ldots, T - 1, \forall I^{Y,t}, d^t\},$$

where $x_{I,x,t,a}^1, y_{I,y,t,d}^1$ are defined recursively by

$$x_{I,x,t,a}^1 = u_{I,x,t,a}^1, \quad \forall a^1;$$

$$x_{I,x,t,a}^t = x_{I,x,t-1}(I^{X,t}, a^{t-1}(I^{X,t})) u_{I,x,t,a}^t, \quad \forall t > 1, I^{X,t}, a^t;$$

$$y_{I,y,t,d}^1 = \beta^0_0 u_{I,y,t,1} r_{I,y,t,1} v_{I,y,t,1} u_{I,y,t,1} r_{I,y,t,1} v_{I,y,t,1}, \quad \forall I^{Y,t}, d^1;$$

$$y_{I,y,t,d}^t = y_{I,y,t-1,d^{t-1}}(I^{Y,t}) u_{I,y,t,d}^t, \quad \forall t > 1, I^{Y,t}, d^t.$$

Then the constraints on $x, y$ are

$$\sum_{a^1} x_{I,x,t,a}^1 = 1; \quad \sum_{a^t} x_{I,x,t,a}^t = x_{I,x,t-1}(I^{X,t}, a^{t-1}(I^{X,t})), \quad \forall t > 1, I^{X,t}; \quad (6.48)$$

$$\sum_{d^1} y_{I,y,t,d}^1 = \beta^0_0 y_{I,y,t,1}(I^{Y,t}), \forall I^{Y,t}; \quad \sum_{d^t} y_{I,y,t,d}^t = y_{I,y,t-1,d^{t-1}}(I^{Y,t}), \forall t > 1, I^{Y,t}. \quad (6.49)$$

Let $X$ be the feasible set of $x$ and $Y(\beta^0)$ be the feasible set of $y$, representing the above constraints on $x, y$, which written them concisely as

$$X := \{x \geq 0|Ex = e\}; \quad Y(\beta^0) := \{y \geq 0|Fy = B\beta^0\}. \quad (6.50)$$

where $E, F, e, B$ are matrices (or vectors) derived from the coefficients of the con-
straint equations, which are similar to the basic model.

Denote \( r_{\delta^T} := (\Pi_{t=1}^{T-1} r_{I_{\delta^T}(\delta^T),s^T(\delta^T)} n_{I_{\delta^T}(\delta^T),a^T(\delta^T)}) r_{I_{\delta^T}(\delta^T),s^T(\delta^T)} \), then the realized probability of a play \( \delta^T \)

\[
\text{Prob}\{\delta^T|x,y\} := x_{I_{\delta^T}(\delta^T),a^T(\delta^T)} y_{I_{\delta^T}(\delta^T),a^T(\delta^T)} r_{I_{\delta^T}(\delta^T),s^T(\delta^T)}.
\]

Therefore the expected payoff under strategies \( x, y \) is

\[
J(x,y) := \sum_{\delta^T} x_{I_{\delta^T}(\delta^T),a^T(\delta^T)} y_{I_{\delta^T}(\delta^T),a^T(\delta^T)} r_{I_{\delta^T}(\delta^T),s^T(\delta^T)} c_{\delta^T}.
\]

(6.51)

where \( c_{\delta^T} \) is the payoff to the attacker when the game ends with \( \delta^T \). Then the problem of finding equilibrium strategies can be formulated as the following optimization problems

\[
\max_{x \in X} \min_{y \in Y(\beta^0)} J(x,y), \quad \min_{y \in Y(\beta^0)} \max_{x \in X} J(x,y),
\]

(6.52)

where \( X, Y(\beta^0) \) are feasible sets of \( x, y \) based on their definitions. Since \( c_{\delta^T} r_{\delta^T} \) for all \( \delta^T \) are known constants, then \( J(x,y) \) is a bilinear function on \( x, y \), the same formulations as in (6.7) and (6.6). Furthermore, the constraints of \( x, y \) in (6.48) and (6.49) represents the same probability conservations as in (6.7) and (6.6). Then theorem 6.1.6 applies, we have that the optimal solutions in (6.52) are the saddle point strategies of the game. Let \( V(\beta^0) \) be the saddle point value in (6.52), By theorem 6.1.8, there is a finite set of vectors \( Q \), such that

\[
V(\beta^0) = \max_{q \in Q} \beta^T q, \quad \forall \beta^0 \in [0,1]^{|S|}.
\]

Next we separate the players' first strategies from their remaining strategies. For
any play $\delta^T$, the realization probability is

\[
\beta_s^0 u_{I_{X,t};a} v_{I_{Y,t};d} n_{I_{Z,t};s} \prod_{t=2}^{T-1} u_{I_{X,t-1;a_t} I_{Z,t};a_t} v_{I_{Y,t};d_t} n_{I_{O,t};o_t} u_{I_{X,t};a_t} I_{Z,t};a_t
\]

where $I_{X,t}, a^t, I_{Y,t}, d^t, I_{Z,t}, s^t, t = 1, \ldots, T$ are all on the path of $\delta^T$. Define the first interaction strategies before $I_{X,2}$ as

\[
x^1 := \{ x_{a_1} | x_{a_1} := u_{I_{X,t};a_1} \forall a_1 \},
\]

\[
y^1 := \{ y_{I_{Y,t};d_1} | y_{I_{Y,t};d_1} := \beta_{0,(I_{Y,t},d_t)}^0 u_{I_{Y,t};d_t} \forall I_{Y,t}, d_t \}
\]

And define the remaining strategies after $I_{X,2}$ as

\[
x^r := \{ x_{I_{X,t};a_t} | \forall I_{X,t}, a_t, t > 1 \};
\]

\[
y^r := \{ y_{I_{Y,t};d_t} | \forall I_{Y,t}, d_t, t > 1 \}
\]

with

\[
x_{I_{X,t};a_t}^r := u_{I_{X,t};a_t} \prod_{t=2}^{T-1} u_{I_{X,t};a_t} I_{Z,t};a_t( I_{X,t} ) \forall I_{X,t}, a_t;
\]

\[
y_{I_{Y,t};d_t}^r := y_{I_{Y,1};d_t} I_{Y,t};d_t \prod_{t=2}^{T-1} u_{I_{Y,t};d_t} I_{Y,t};d_t( I_{Y,t} ) \forall I_{Y,t}, d_t, t \geq 2
\]

Based on their definition, the probability constraints on $x^1, y^1, x^r, y^r$ are

\[
\sum_{a_1} x_{a_1}^1 = 1;
\]

\[
\sum_{d_1} y_{I_{Y,1};d_1} = \beta_{0,(I_{Y,1})}^0 \forall I_{Y,1};
\]

\[
\sum_{d_2} y_{I_{Y,2},d_2} = y_{I_{Y,1};d_2}( I_{Y,2} ) \forall I_{Y,2};
\]

\[
\sum_{a_2} x_{I_{X,t},a_2}^1 = 1, \forall I_{X,2};
\]

\[
\sum_{a_t} x_{I_{X,t},a_t}^r = x_{I_{X,t-1};a_t-1}( I_{X,t} ), \forall I_{X,t}, t > 2;
\]

\[
\sum_{d_t} y_{I_{Y,t};d_t}^r = y_{I_{Y,t-1};d_t-1}( I_{Y,t} ), \forall I_{Y,t}, t > 2.
\]

Denote the feasible set for $x^1, x^r$ as $X^1, X^r$ respectively. Notice the constraints of $y^r$
depend on $y^1$ and the constraints of $y^1$ depend on $\beta^0$, denote the feasible set of $y^r$ as $Y^r(y^1)$ and the feasible set of $y^1$ as $Y^1(\beta^0)$.

Denote $r^T_{\delta^T} := (\prod_{t=2}^{T-1} r_{I(z,t)}(\delta^T), s^1(\delta^T), n_{I(z,t)}(\delta^T), o^1(\delta^T)) r_{I(z, T)}(\delta^T), s^T(\delta^T)$, then the realization probability of $\delta^T$ can be written as

$$
\sum_{\delta^T} x^1_{\alpha^1(\delta^T), r^T_{I(z,1)}(\delta^T), s^1(\delta^T), n_{I(z,1)}(\delta^T), o^1(\delta^T)} x^r_{I(z, T), o^T(\delta^T), s^T(\delta^T), n_{I(z, T)}(\delta^T), o^T(\delta^T)} r^T_{\delta^T} c_{\delta^T}.
$$

Given $x^1, y^1, x^r, y^r$ and $\beta^0$, the expected payoff is

$$
\tilde{J}(x^1, y^1, x^r, y^r) = \sum_{\delta^T} x^1_{\alpha^1(\delta^T), r^T_{I(z,1)}(\delta^T), s^1(\delta^T), n_{I(z,1)}(\delta^T), o^1(\delta^T)} x^r_{I(z, T), o^T(\delta^T), s^T(\delta^T), n_{I(z, T)}(\delta^T), o^T(\delta^T)} \sum_{\delta^T \in I(z,2)} x^r_{I(z, T), o^T(\delta^T), s^T(\delta^T), n_{I(z, T)}(\delta^T), o^T(\delta^T)} r^T_{\delta^T} c_{\delta^T},
$$

where the second sum groups plays $\delta^T$ into the different information sets $I(z,2)$ that they belong to. By the definition of $x^1, x^r$, maximizing $x$ in $J(x, y)$ is equivalent to maximizing $x^1, x^r$ in $\tilde{J}(x^1, y^1, x^r, y^r)$. Similarly by the definition of $y^1, y^r$, minimizing $y$ in $J(x, y)$ is equivalent to minimizing $y^1, y^r$ in $\tilde{J}(x^1, y^1, x^r, y^r)$, therefore (6.52) is equal to

$$
J_{MM}: = \max_{x^1 \in X^1} \min_{y^1 \in Y^1(\beta^0)} J(x^1, y^1, x^r, y^r),
$$

$$
J_{mm}: = \min_{y^1 \in Y^1(\beta^0)} \max_{x^1 \in X^1} J(x^1, y^1, x^r, y^r).
$$

Notice $V(\beta^0)$ is defined as the optimal value of (6.52), then we have

$$
J_{MM} = J_{mm} = V(\beta^0).
$$
Consider the following optimization problems

\[
J_{mMMm} := \min_{y^1 \in Y^1} \max_{x^1 \in X^1} \min_{y^r \in Y^r(y^1)} \max_{x^r \in X^r} \tilde{J}(x^1, y^1, x^r, y^r);
\]  \hspace{1cm} (6.55)

\[
J_{MmMm} := \max_{x^1 \in X^1} \min_{y^1 \in Y^1} \max_{x^r \in X^r} \min_{y^r \in Y^r(y^1)} \tilde{J}(x^1, y^1, x^r, y^r),
\]  \hspace{1cm} (6.56)

where (6.55) is the result of exchanging the optimization orders of \(x^1\) and \(y^r\) in (6.54) and (6.56) is the result of exchanging the optimization orders of \(y^1\) and \(x^r\) in (6.53). With the same deduction as that for the basic model, we can show that

**Theorem 6.3.2** The saddle point strategies \(u^1^*, y^{1^*}\) in (6.56) and (6.55) are saddle point strategies of the first interaction in the original game.

Therefore we can apply the subgame decomposition method for this game as for the basic model.
Chapter 7

Games with Markov structure

In chapter 6, we have developed a subgame decomposition method which exploits the nested information structure and decomposes the multi-stage game into a sequence of one-stage subgames. We show that the equilibrium strategies in the original game can be found by solving these subgames. In this method, we need to estimate a value function at each information set of the less-informed player. The number of these information sets and their sizes grow exponentially with the number of stages. Therefore, the subgame decomposition method requires lots of computation. However, for games with Markov structure, we will show the method just needs to estimate $T-1$ value functions with a constant number of variables. Therefore we can reduce computation exponentially.

**Definition 7.0.3** A dynamic game has Markov structure if

- **Markov transition probability** The transition probability to a new state $s^t$ completely depends on current state $s^{t-1}$, current actions $a^t, d^t$, i.e.,

  \[ T[\tilde{a}^{t-1}, a^t, d^t], s^t = T[\tilde{a}^{t-1}, a^t, d^t], s^t, \forall s^t, \text{ if } s^{t-1}(\tilde{a}^{t-1}) = s^{t-1}(\tilde{a}^{t-1}). \]

- **Additive stage costs** The total payoff is a sum of per stage payoffs, each of them only depends on current state $s^{t-1}$, current actions $a^t, d^t$, i.e., there is a set of functions $c_t(s^{t-1}, a^t, d^t), t = 1, \ldots, T$ such that

  \[ c_{a^t} = \sum_{t=1}^{T} c_t(s^{t-1}(\sigma^T), a^t(\sigma^T), d^t(\sigma^T)), \forall \sigma^T. \]
• **Constant per stage actions sets** Any information sets at the same level have the same set of actions. That is, for a fixed stage \( t \) and a fixed player \( n, n = X, Y, Z \), any two information sets \( I_{n,t}^s, \tilde{I}_{n,t}^s \) satisfy \( A(I_{n,t}^s) = A(\tilde{I}_{n,t}^s) \).

For games with Markov structure, we will show that there is a common cost-to-go function for each stage, which only depends on the conditional distribution on the information set \( I_{X,t}^X \) at that level. Instead of estimating the payoff functions for all the subgames at that stage, one just needs to estimate this cost-to-go function. We will show these results for \( t = 2 \). With recursive decomposition, one can generalize them for the subgames in other levels.

**Definition 7.0.4** For two sequences \( I^L, I^S \), the remainder of \( I^L \) minus \( I^S \) is defined as the subsequence of \( I^L \) that consists all the elements NOT in \( I^S \), denoted as \( I^L - I^S \).

For example, \( \sigma^1 = [s^0, a^1, d^1], \sigma^2 = [s^0, a^1, d^1, s^1, a^2, d^2] \), then \( \sigma^2 - \sigma^1 = [s^1, a^2, d^2] \). Another example, \( \sigma^2 = [s^0, a^1, d^1], I_{X,2}^X = [a^1, d^1] \), then \( \sigma^2 - I_{X,2}^X = s^0 \).

### 7.1 The Common “Cost-to-go” Problems

As discussed in the previous chapter, given the first stage saddle point strategies \( u^1, y^1 \), the optimization problems for subgame \( G(I_{X,2}^X, \cdot) \) are

\[
\begin{align*}
\max_{x^{I_{X,2}^X} \in X_{I_{X,2}^X}} \min_{y^{I_{X,2}^Y} \in Y_{I_{X,2}^Y}(x^{I_{X,2}^X}(x^1,y^1))} J_{I_{X,2}}^r(x^{I_{X,2}^X}, y^{I_{X,2}^Y}); \\
\min_{y^{I_{X,2}^Y} \in Y_{I_{X,2}^Y}(x^{I_{X,2}^X}(x^1,y^1))} \max_{x^{I_{X,2}^X} \in X_{I_{X,2}^X}} J_{I_{X,2}}^r(x^{I_{X,2}^X}, y^{I_{X,2}^Y}),
\end{align*}
\]  

(7.1) (7.2)

with the objective function \( J_{I_{X,2}}^r(x^{I_{X,2}^X}, y^{I_{X,2}^Y}) \) as

\[
J_{I_{X,2}}^r(x^{I_{X,2}^X}, y^{I_{X,2}^Y}) := \sum_{\sigma^1, \sigma^2} \sum_{x_{I_{X,2}^X}(\sigma^1, \sigma^2), y_{I_{X,2}^Y}(\sigma^2)} I_{X,2}^{I_{X,2}^X} d_{I_{X,2}^X}(\sigma^1) c_{I_{X,2}^Y}(\sigma^2),
\]

(7.3)
where $r_s^T := \prod_{t=2}^{T} r_{I^T,s,t}$. The probability constraints on $x_{X,2}^{I,2}, y_{Y,2}^{I,2}$ is

\begin{align}
\sum_{a^2} x_{I,X,2,a^2}^{I,2} = 1; \sum_{d^2} y_{I,Y,2,d^2}^{I,2} &= \beta^{I,X,2} (x^1, y^1)_{(Y,2)}, \forall Y,2 \rightarrow I^{X,2}; \quad (7.4) \\
\sum_{a^t} x_{I,X,t,a^t}^{I,2} = x_{I,X,t-1(I,X,t),a^t-1(I,X,t)}, \forall I^{X,t}, t > 2; \quad (7.5) \\
\sum_{d^t} y_{I,Y,t,d^t}^{I,2} = y_{I,Y,t-1(I,Y,t),d^t-1(I,Y,t)}, \forall I^{Y,t}, t > 2. \quad (7.6)
\end{align}

Lemma 7.1.1 In a game with Markov structure, given two player X's information sets, $\hat{I}_{X,2} := [\hat{a}^1, \hat{d}^2], \bar{I}_{X,2} := [\bar{a}^1, \bar{d}^2]$, for any sequences $\hat{I}^{Y,t}, \hat{\sigma}^T, \hat{\sigma}^1 \rightarrow \hat{I}_{X,2}$, there must be another sequences $\hat{I}^{Y,t}, \hat{\sigma}^T, \hat{\sigma}^1 \rightarrow \bar{I}_{X,2}$ such that

\begin{align*}
\hat{I}^{Y,t} - [\hat{a}^1, \hat{d}^2] &= \bar{I}^{Y,t} - [\bar{a}^1, \bar{d}^2]; \\
\hat{\sigma}^1 - [\hat{a}^1, \hat{d}^2] &= \bar{\sigma}^1 - [\bar{a}^1, \bar{d}^2]; \\
\hat{\sigma}^T - [\hat{a}^1, \hat{d}^2] &= \bar{\sigma}^T - [\bar{a}^1, \bar{d}^2];
\end{align*}

denoted as $\hat{I}^{Y,t} \leftrightarrow \bar{I}^{Y,t}; \hat{\sigma}^T \leftrightarrow \bar{\sigma}^T; \hat{\sigma}^1 \leftrightarrow \bar{\sigma}^1$ respectively.

Proof: Since $\hat{I}^{Y,t}, \hat{\sigma}^1$ are all subsets of $\hat{\sigma}^T$, we just need to show for any play

\begin{align*}
\hat{\sigma}^T := [\hat{s}^0, \hat{a}^1, \hat{d}^1, \ldots, \hat{d}^{T-1}, \hat{a}^T, \hat{s}^T] \rightarrow \hat{I}^{X,2},
\end{align*}

there must be another play

\begin{align*}
\bar{\sigma}^T := [\bar{s}^0, \bar{a}^1, \bar{d}^1, \ldots, \bar{d}^{T-1}, \bar{a}^T, \bar{s}^T] \rightarrow \bar{I}^{X,2}.
\end{align*}

Since $\hat{s}^0$ is the initial action, if $\hat{\sigma}^T$ and $\hat{I}^{X,2}$ exist, then $\bar{I}^{Z,1} := [\bar{s}^0, \bar{a}^1, \bar{d}^1]$ must exist. Because the property of Constant per stage actions sets in the Markov game, $\bar{I}^{Z,1}$ has the same set of action as $[\hat{s}^0, \hat{a}^1, \hat{d}^1]$, therefore, $[\hat{s}^0, \hat{a}^1, \hat{d}^1, \hat{\sigma}^1]$ must exist. With similar deduction adding element one by one, we show that $\bar{\sigma}^T$ exists. $\Box$

Consider the constraints of (7.4) to (7.6) for two subgames starting at different information sets $\hat{I}^{X,2}, \bar{I}^{X,2}$, map the variables $x_{I,X,t,a^t}, y_{I,Y,t,d^t}$ in subgame $G(I^{X,2}, \beta^{I,X,2})$
with the variables $x^{I_{X,2}}_{I_{X,T},a_T}, y^{I_{X,2}}_{I_{T,Y,T},d_T}$ in subgame $G(\tilde{I}^{X,2}, \beta^{I^{X,2}})$ based on $\hat{I}^{X,t} = \hat{I}^{X,t}; \tilde{I}^{Y,t}$ then if $\beta^{I^{X,2}} = \beta^{I^{X,2}}$, they’re essentially the same constraints. Lemma 7.1.2 summarizes the above discussion.

**Lemma 7.1.2** In a Markov game, any two subgames $G(\tilde{I}^{X,2}, \beta^{I^{X,2}}), G(\tilde{I}^{X,2}, \beta^{I^{X,2}})$ have the same feasible sets on $x^{I_{X,2}}$. They also have the same feasible sets on $y^{I_{X,2}}$ if they have the same initial distribution $\beta^{I^{X,2}} = \beta^{I^{X,2}}$.

Due to the property of additive stage costs, we can define

$$c_{\sigma}^{1} := c^{1}(s^{0}(\sigma^{1}), a^{1}(\sigma^{1}), d^{1}(\sigma^{1})); \quad c_{\sigma}^{r_{T}} := \sum_{t=2}^{T} c^{t}(s^{t-1}(\sigma^{T}), a^{t}(\sigma^{T}), d^{t}(\sigma^{T})).$$

Then for any $\sigma^{T}$, $c_{\sigma}^{r_{T}} = c_{\sigma}^{1} + c_{\sigma}^{r_{T}}$, where $c_{\sigma}^{r_{T}}$ only depends on $\sigma^{T} - I^{X,2}$. Substitute $c_{\sigma}^{r_{T}}$ with $c_{\sigma}^{1} + c_{\sigma}^{r_{T}}$ in $J^{r_{I_{X,2}}}(x^{I_{X,2}}, y^{I_{X,2}})$, we have

$$J^{r_{I_{X,2}}}(x^{I_{X,2}}, y^{I_{X,2}})$$

$$= \sum_{\sigma^{T} \succ I^{X,2}} x^{I_{X,2}}_{I_{X,T}(\sigma^{T}),a^{T}(\sigma^{T})} y^{I_{X,2}}_{I_{T,Y,T}(\sigma^{T}),d^{T}(\sigma^{T})} c_{\sigma^{T}}^{r_{T}}$$

$$= \sum_{\sigma^{1} \succ I^{X,2}} \sum_{\sigma^{T} \succ \sigma^{1}} x^{I_{X,2}}_{I_{X,T}(\sigma^{T}),a^{T}(\sigma^{T})} y^{I_{X,2}}_{I_{T,Y,T}(\sigma^{T}),d^{T}(\sigma^{T})} (c_{\sigma^{1}}^{1}(\sigma^{T}) + c_{\sigma^{T}}^{r_{T}})$$

$$= \sum_{\sigma^{1} \succ I^{X,2}} c_{\sigma^{1}}^{1}(\sum_{\sigma^{T} \succ \sigma^{1}} x^{I_{X,2}}_{I_{X,T}(\sigma^{T}),a^{T}(\sigma^{T})} y^{I_{X,2}}_{I_{T,Y,T}(\sigma^{T}),d^{T}(\sigma^{T})} c_{\sigma^{T}}^{r_{T}})$$

$$+ \sum_{\sigma^{T} \succ I^{X,2}} x^{I_{X,2}}_{I_{X,T}(\sigma^{T}),a^{T}(\sigma^{T})} y^{I_{X,2}}_{I_{T,Y,T}(\sigma^{T}),d^{T}(\sigma^{T})} c_{\sigma^{T}}^{r_{T}}.$$


Because

\[
\sum_{\sigma T \succ \sigma 1} x_{I X, T(\sigma T), a T(\sigma T)} x_{I Y, T(\sigma T), d T(\sigma T)} x_{I T(\sigma T), s T(\sigma T)}
= \beta_{\sigma 1} x_{I X, 2} \sum_{\sigma T \succ \sigma 1} \prod_{t=2}^{T} u_{I X, t(\sigma T), a t(\sigma T)} u_{I Y, t(\sigma T), d t(\sigma T)} u_{I T, t(\sigma T), s t(\sigma T)}
= \beta_{\sigma 1} x_{I X, 2} \sum_{\sigma T \succ \sigma 1} \prod_{t=2}^{T-1} u_{I X, t(\sigma T-1), a t(\sigma T-1)} u_{I Y, t(\sigma T-1), d t(\sigma T-1)} u_{I T, t(\sigma T-1), s t(\sigma T-1)}
\]

Therefore \( J_{I X, 2}(x_{I X, 2}, y_{I X, 2}) \) is equal to

\[
\sum_{\sigma 1 \succ I X, 2} \beta_{\sigma 1} x_{I X, 2} c_{\sigma 1} + \sum_{\sigma T \succ I X, 2} x_{I X, T(\sigma T), a T(\sigma T)} y_{I Y, T(\sigma T), d T(\sigma T)} c_{T} r_{T} \sigma T.
\]

Then (7.1) and (7.2) can be written as

\[
\sum_{\sigma 1 \succ I X, 2} \beta_{\sigma 1} x_{I X, 2}(x^{1}, y^{1}) c_{\sigma 1} + \]

\[
\max_{x^{I X, 2} \in X^{I X, 2}, y^{I X, 2} \in Y^{I X, 2}} \min_{\beta_{I X, 2}(x^{1}, y^{1})} \sum_{\sigma T \succ I X, 2} x_{I X, T(\sigma T), a T(\sigma T)} y_{I Y, T(\sigma T), d T(\sigma T)} c_{T} r_{T} \sigma T.
\]

\[
\sum_{\sigma 1 \succ I X, 2} \beta_{\sigma 1} x_{I X, 2}(x^{1}, y^{1}) c_{\sigma 1} + \]

\[
\min_{y^{I X, 2} \in Y^{I X, 2}} \max_{\beta_{I X, 2}(x^{1}, y^{1})} \sum_{\sigma T \succ I X, 2} x_{I X, T(\sigma T), a T(\sigma T)} y_{I Y, T(\sigma T), d T(\sigma T)} c_{T} r_{T} \sigma T.
\]
and they have the same saddle point strategies as the following problems:

\[
\begin{align*}
\max_{x^{I_X,2} \in X^{I_X,2}, y^{I_X,2} \in Y^{I_X,2}(\beta^{I_X,2}, x^{I_X,2})} & \quad \min_{y^{I_X,2} \in Y^{I_X,2}} \quad \sum_{\sigma^{T} \in I^{I_X,2}} x_{I^{I_X,2}}^{I_X,2}(\sigma^{T}) a^{I_X,2}(\sigma^{T}) y_{I^{I_X,2}}^{I_X,2}(\sigma^{T}) d^{I_X,2}(\sigma^{T}) c^{I_X,2} \sigma^{T} r^{I_X,2} \sigma^{T}; \\
\min_{y^{I_X,2} \in Y^{I_X,2}(\beta^{I_X,2}, x^{I_X,2})} & \quad \max_{x^{I_X,2} \in X^{I_X,2}} \quad \sum_{\sigma^{T} \in I^{I_X,2}} x_{I^{I_X,2}}^{I_X,2}(\sigma^{T}) a^{I_X,2}(\sigma^{T}) y_{I^{I_X,2}}^{I_X,2}(\sigma^{T}) d^{I_X,2}(\sigma^{T}) c^{I_X,2} \sigma^{T} r^{I_X,2} \sigma^{T}.
\end{align*}
\]

By lemma 7.1.2, for subgames that have the same \( \beta^{I_X,2} \), the corresponding feasible sets of \( x^{I_X,2}, y^{I_X,2} \) are the same. Furthermore, because additive stage costs and Markov transition probability, \( c^{I_X,2} = \sigma^{T} r^{I_X,2} \sigma^{T} \). Therefore, if \( \beta^{I_X,2} = \beta^{I_X,2} \), the optimization problems (7.7) and (7.8) are the same for both subgames \( G(I^{I_X,2}, \beta^{I_X,2}), G(I^{I_X,2}, \beta^{I_X,2}) \). Then we have

**Theorem 7.1.3** In a Markov game, any two subgames \( G(I^{I_X,2}, \beta^{I_X,2}), G(I^{I_X,2}, \beta^{I_X,2}) \) have the same optimization problems (7.7) and (7.8) if \( \beta^{I_X,2} = \beta^{I_X,2} \).

By theorems of 6.1.6 and 6.1.8, (7.7) and (7.8) have the same saddle point value. As a function of the initial distribution \( \beta^{I_X,2} \), this saddle point value can be represented by a finite set of support vectors. Since this "cost-to-go" game does not depend on any specific information set \( I^{I_X,2} \) at that level, without referring to a specific information set we can just denote it as \( G^2(\beta^1) \) with \( \beta^1 \) as the distribution of the initial states in the "cost-to-go" game. The corresponding saddle point value function is denoted as \( V^2(\beta^1) \). By theorem 7.1.3, instead of estimating the saddle point value function \( V^{I_X,2}(\beta^{I_X,2}) \) for each subgame \( G(I^{I_X,2}, \cdot) \), one can estimate \( V^2(\beta^1) \) for \( G^2(\beta^1) \).

Notice \( G^2(\beta^1) \) can also be taken as a "cost-to-go" game starting on an information set \( I^{I_X,2} \) without considering the cost of the first interaction \( c^1(s^0, a^1, d^1) \). One can further decompose \( G^2(\beta^1) \) and find the "cost-to-go" game for the next level subgames. By doing this iteratively, one "cost-to-go" game for one level, we just need to estimate \( T - 1 \) saddle point value functions, \( V^t(\beta^{t-1}), t = 2, \ldots, T \), where \( T \) is the number of interactions in the original game.
Figure 7.1: Grouping nodes (within the circles) with the same latest state to reduce the sizes of subgames.

7.2 The Indifferent Histories

Within a subgame $G(I^{X,t}, \beta^{X,t})$, the histories $\sigma^{t-1}$ within its initial information set $I^{X,t}$ can be written as

$$\sigma^{t-1} = [s^0, a^1, d^1, s^1, \ldots, a^{t-1}, d^{t-1}, s^{t-1}].$$

Notice $I^{X,t} = [a^1, d^1, \ldots, a^{t-1}, d^{t-1}]$, then the number of histories within that information set (histories with the same subsequence $I^{X,t}$ but different subsequence $\sigma^{t-1} - I^{X,t}$) grows exponentially with the number of stages $t$. However, for games with Markov structure, we will “group” the histories that share the same current state $s^{t-1}$ into one “node”, as shown in Fig. 7.1. In this approach, the size of $I^{X,t}$ is the number of possible underlying states $|S|$, which is constant over time. In following discussion, without specific instruction, all sequences are restricted to be the successors of a given $I^{X,t}$.

Lemma 7.2.1 In a Markov game, given two histories $\hat{\sigma}^1 := [\hat{s}^0, a^1, d^1, \hat{s}^1]$, $\tilde{\sigma}^1 := [\tilde{s}^0, a^1, d^1, \tilde{s}^1]$, for any sequences $\hat{Y}_{t}, \hat{T} \succeq \hat{\sigma}^1$, there must be another sequences $\tilde{Y}_{t}, \tilde{T} \succeq \tilde{\sigma}^1$. 
such that
\[ I^{Y,t} - \tilde{\sigma}^1 = \tilde{I}^{Y,t} - \tilde{\sigma}^1; \quad \sigma^T - \tilde{\sigma}^1 = \tilde{\sigma}^T - \tilde{\sigma}^1; \]
denoted as \[ \tilde{I}^{Y,t} \xrightarrow{\sigma^1} I^{Y,t}; \quad \tilde{\sigma}^T \xrightarrow{\sigma^1} \tilde{\sigma}^T \] respectively where \( \sigma^1 \) is the space of \( \tilde{\sigma}^1, \tilde{\sigma}^1 \).

**Proof:** Since \( I^{Y,t} \) is a subset of \( \sigma^T \), we just need to show for any play
\[ \tilde{\sigma}^T := [s^0, a^1, d^1, \hat{s}_1, \ldots, d^{T-1}, \hat{a}^T, \hat{s}^T] \succ \tilde{\sigma} \]
there must be another play
\[ \tilde{\sigma}^T := [s^0, a^1, d^1, \hat{s}_1, \ldots, d^{T-1}, \hat{a}^T, \hat{s}^T] \succ \tilde{\sigma}^1. \]

Because the property of *Constant per stage actions sets* in the Markov game, on \( \tilde{\sigma}^1, \tilde{\sigma}^1 \), Nature has the same set of actions \( s^2 \). Therefore the existence of \( [\tilde{\sigma}^1, \hat{s}^2] \) indicates the existence of \( [\tilde{\sigma}^1, \hat{s}^2] \). With similar deduction adding element one by one down to a play \( \sigma^T \). We prove the lemma.

By lemma 7.2.1, player Y's behavior strategies on information sets \( \tilde{I}^{Y,t} \succ \tilde{\sigma}^t \) can be applied to the information sets \( \tilde{I}^{Y,t} \succ \tilde{\sigma}^t \) if \( \tilde{I}^{Y,t} \xrightarrow{\sigma^1} \tilde{I}^{Y,t} \). Then grouping \( \tilde{\sigma}^1, \tilde{\sigma}^t \) means restricting player Y's behavior strategies after \( \tilde{\sigma}^t, \tilde{\sigma}^t \), such that
\[ \psi_{\tilde{I}^{Y,t}, \tilde{\sigma}^t} = \psi_{\tilde{I}^{Y,t}, \tilde{\sigma}^t} \text{ if } \tilde{I}^{Y,t} \xrightarrow{\sigma^1} \tilde{I}^{Y,t}. \]

We will show that for two \( \tilde{\sigma}^1, \tilde{\sigma}^1 \) in the same \( I^{X,2} \) with \( s^1(\tilde{\sigma}^1) = s^1(\hat{\sigma}^1) \), grouping \( \tilde{\sigma}^1, \tilde{\sigma}^1 \) does not change the saddle point value, thus player Y can use the same strategies on these histories and their matched successors (with relationship \( \sigma^1 \)) without getting worse payoff.

Consider a "cost-to-go" game \( G^2(\beta^1) \) with player X's initial information set as \( I^{X,2} \). Given the behavior strategies \( u, v \) on the game. We can calculate the expected
payoff for any node $\sigma^1 \succ I^{X,2}$ as
\[
p(\sigma^1) := \sum_{\sigma^t \succ \sigma^1} \prod_{t=2}^{T} u_{I^{X,\tau}(\sigma^T),a^T(\sigma^T),d^T(\sigma^T)} I^{Y,\tau}(\sigma^T),s^T(\sigma^T) r_{\sigma^T}; \tag{7.9}
\]
where $c_{\sigma^T}^1 := \sum_{t=2}^{T} c^r(s^{T-1}(\sigma^T),a^T(\sigma^T),d^T(\sigma^T))$. Let $\hat{\sigma}^1, \bar{\sigma}^1$ be two histories within $I^{X,2}$ such that $s^1(\hat{\sigma}^1) = s^1(\bar{\sigma}^1)$. By lemma 7.2.1, for any history $\hat{\sigma}^T \succ \bar{\sigma}^1$, there must be another history $\bar{\sigma}^T \succ \hat{\sigma}^1$, such that $\hat{\sigma}^T \xrightarrow{\sigma^1} \bar{\sigma}^T$. Therefore, for any $\tau > 1$,
\[
I^{X,\tau}(\hat{\sigma}^T) = I^{X,\tau}(\bar{\sigma}^T); \quad a^{\tau}(\hat{\sigma}^T) = a^{\tau}(\bar{\sigma}^T); \quad d^{\tau}(\hat{\sigma}^T) = d^{\tau}(\bar{\sigma}^T); \quad s^{\tau}(\hat{\sigma}^T) = s^{\tau}(\bar{\sigma}^T).
\]
Thus $u_{I^{X,\tau}(\hat{\sigma}^T),a^{\tau}(\sigma^T)} = u_{I^{X,\tau}(\bar{\sigma}^T),a^{\tau}(\sigma^T)}; \quad c_{\hat{\sigma}^T}^1 = c_{\bar{\sigma}^T}^1$. Furthermore, by the property of Markov Transition Probability, we have
\[
r_{I^{Z,\tau}(\hat{\sigma}^T),s^{\tau}(\sigma^T)} = r_{I^{Z,\tau}(\bar{\sigma}^T),s^{\tau}(\sigma^T)}, \forall \tau > 1.
\]
As a result, the difference between $p(\hat{\sigma}^1)$ and $p(\bar{\sigma}^1)$ totally depends on player Y's strategy $v$. The following lemma summarizes the above result.

Lemma 7.2.2 In $G^2(\beta^1)$, let $\hat{\sigma}^1, \bar{\sigma}^1$ be two histories within $I^{X,2}$ such that $s^1(\hat{\sigma}^1) = s^1(\bar{\sigma}^1)$, then the difference of their expected payoff values totally depends on player Y's strategy $v$. That is, if
\[
u_{I^{Y,\tau}(\hat{\sigma}^T),d^{\tau}(\sigma^T)} = \nu_{I^{Y,\tau}(\bar{\sigma}^T),d^{\tau}(\sigma^T)}, \forall \tau > 1
\]
then $p(\hat{\sigma}^1) = p(\bar{\sigma}^1)$.

Lemma 7.2.3 Denote the saddle point strategies of $G^2(\beta^1)$ as $u^*, v^*$. Let $p^*(\sigma^1)$ be the expected payoff on node $\sigma^1$ given $u^*, v^*$. Let $\hat{\sigma}^1, \bar{\sigma}^1$ be two histories within $I^{X,2}$ such that $s^1(\hat{\sigma}^1) = s^1(\bar{\sigma}^1)$. If $\beta^1_{\alpha^1}, \beta^{\gamma^1} > 0$, then $p^*(\hat{\sigma}^1) = p^*(\bar{\sigma}^1)$.

Proof: Assume $p^*(\hat{\sigma}^1) > p^*(\bar{\sigma}^1)$, then player Y can change its strategies such that $v_{I^{Y,\tau},d^\tau} \leftarrow v_{I^{Y,\tau},d^\tau}$ if $\hat{Y}^{\tau,\sigma^1} \xrightarrow{\sigma^1} \bar{Y}^{\tau,\sigma^1}$. By lemma 7.2.2, the difference of expected values on $\hat{\sigma}^1$ and $\bar{\sigma}^1$ totally depends on player Y's strategies. Notice after that change,
these strategies are the same, then the new expected payoff on $\hat{\sigma}^1$ should be equal to $p^*(\hat{\sigma}^1)$. Since $\beta_{o^1} > 0$, player $Y$ can decrease the expected payoff to player $X$ by the amount of $\beta_{o^1}(p^*(\hat{\sigma}^1) - p^*(\tilde{\sigma}^1))$, which contradicts with $v^*$ is the saddle point strategy. Therefore, the assumption is false. With the same argument, one can rule out $p^*(\hat{\sigma}^1) < p^*(\tilde{\sigma}^1)$. As a result, $p^*(\hat{\sigma}^1) = p^*(\tilde{\sigma}^1)$.

Assume we have the saddle point strategies $u^*, v^*$ for the subgame $G^2(\beta^1)$. For any histories within the initial information set $I^{X,2}$ of the game that share the same current state $s^1$, by lemma 7.2.3, player $Y$ can make these strategies to be the same without getting worse payoff. As discussed before, these histories are naturally indifferent to player $X$. Therefore, we have

**Theorem 7.2.4** In a "cost-to-go" subgame $G^2(\beta^1)$, by restricting player $Y$'s strategy on any histories in its initial information set that have the same latest underlying state $s^1$ (and their successors with mapping $\sigma^1 \mapsto s^1$), player $Y$ does not get worse payoff.

By theorem 7.2.4, any histories in $I^{X,2}$ in the subgame $G^2(\beta^1)$ can be grouped according to $s^1$. Specifically, for any $\hat{s}^1$ we merge $\{\sigma^1 | s^1(\sigma^1) = \hat{s}^1\}$ as a node. The corresponding distribution is changed to $\beta^1 := \{\beta_{s^1}\beta_{o^1} := \sum_{s^1 \rightarrow \beta} \beta_{s^1}, \forall s^1\}$. After merging, the number of nodes within $I^{X,2}$ is $|S|$. One can repeat the same process in subgame $G^{t+1}(\beta^t)$ for $t > 2$ and keep the number of nodes within $I^{X,t+1}$ constantly being $|S|$. Combined with the result of the common "cost-to-go" subgame, the computation for games with Markov structure is reduced exponentially.

### 7.3 Numerical Results

In this section we use our method to solve multi-stage network interdiction games with nested-information and compare with the method (called Large LP) in (von Stengel, 1996), which solves the bilinear problems (6.6), (6.7) of the multi-stage game. Numerical results show that our method is about 2 to 3 orders of magnitude faster.
Figure 7.2: The underlying networks for the nested information game. For each arc, the first number is the capacity, and the second number (if any) is the probability of broken if attacked. Arcs without this number are not interdictable.

More important, when the problem size increases and the Large LP fails to output solution, our method can still solve the problem within a short time.

Consider an interdiction game on the network shown in Fig. 7.2, where at the beginning each interdictable arc exists with probability of 50%, which are independent across arcs. At each stage the attack randomly selects an arc to attack. After observing the attacker's action, the defender decides whether to send flows on the attacked arc if it's available. If it chooses not to send flow, i.e. to deceive the attacker, the defender needs to pay the deception cost as $1/10$ of the difference between the maximum flow when it uses the attacked arc and the maximum flow when it does not use this arc. At the final stage, the payoff to the defender is the max-flow of the resulting network, i.e., the defender's cost is the negative of the maximum flow. The attacker is to maximize the defender's total cost, the sum of the costs in all stages. The network state depends on the previous network state as well as players' current actions. As a result, the game has Markov structure.

The number of attackable arcs and the number of interactions can inflate the
Table 7.1: The number of states and the number of players’ actions grow with the number of attackable arcs ($A$), assuming the game has constant action sets over time.

<table>
<thead>
<tr>
<th>$A$</th>
<th>States $X$ actions</th>
<th>$X$ actions</th>
<th>$Y$ actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>$2^3 = 8$</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>$2^4 = 16$</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>$2^5 = 32$</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>$2^6 = 64$</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Problem size exponentially. Table 7.1 shows that the possible states of the network grow exponentially with the number of attackable arcs. Consider the LP problem (6.14)

\[
\max_{x \geq 0, Ex = e} \min_{y \geq 0, Fy = f} x'Cy.
\]

In the case where the number of actions are constant over time, let $n_a, n_d, n_s$ be the number of actions for player $X$, player $Y$ and Nature respectively. Denote $t$ as the total number of interactions in the game. Then the size of $E, F, C$ have the following relationship with $n_a, n_d, n_s$ and $t$

\[
(E_r^1, E_c^1) = (1, n_a); (F_r^1, F_c^1) = (n_a * n_s, n_a * n_d * n_s); (C_r^1, C_c^1) = (E_c^1, F_c^1);
\]

\[
(E_r^{t+1}, E_c^{t+1}) = (E_r^t * n_a * n_d + 1, (E_c^t n_d + 1) * n_a),
\]

\[
(F_r^{t+1}, F_c^{t+1}) = ((F_r^t * n_d + 1) * n_a * n_s, (F_c^t + 1) * n_a n_d * n_s),
\]

\[
(C_r^{t+1}, C_c^{t+1}) = ((E_c^t n_d + 1) * n_a, (F_c^t + 1) * n_a n_d * n_s), \forall t \geq 0;
\]

where $M_r, M_c$ are the row number and the column number of the matrix $M$ with $M = E, F, C$. Table 7.2 shows that the sizes of $x, y, C, E, F$ grow exponentially with the number of interactions $T$ and the number of attackable arcs.

The running times (in terms of seconds, if not specified) are shown in Table 7.3. The second column is the number of support vectors of the expected payoff functions.
Table 7.2: The size of optimization problems grows exponentially with the number of interactions (T).

<table>
<thead>
<tr>
<th>(A, T)</th>
<th>x</th>
<th>y</th>
<th>E</th>
<th>F</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3, 1)</td>
<td>3</td>
<td>48</td>
<td>1 x 3</td>
<td>24 x 48</td>
<td>3 x 48</td>
</tr>
<tr>
<td>(3, 2)</td>
<td>21</td>
<td>2352</td>
<td>7 x 21</td>
<td>1176 x 2352</td>
<td>21 x 2352</td>
</tr>
<tr>
<td>(3, 3)</td>
<td>129</td>
<td>112944</td>
<td>43 x 129</td>
<td>56472 x 112944</td>
<td>129 x 112944</td>
</tr>
<tr>
<td>(4, 1)</td>
<td>4</td>
<td>128</td>
<td>1 x 4</td>
<td>64 x 128</td>
<td>4 x 128</td>
</tr>
<tr>
<td>(4, 2)</td>
<td>26</td>
<td>16512</td>
<td>9 x 36</td>
<td>8256 x 16512</td>
<td>36 x 16512</td>
</tr>
<tr>
<td>(4, 3)</td>
<td>292</td>
<td>2113664</td>
<td>73 x 292</td>
<td>1056832 x 2113664</td>
<td>292 x 2113664</td>
</tr>
</tbody>
</table>

Table 7.3: Compare subgame decomposition method and the Large LP method in (von Stengel, 1996). T is the number of interactions and A is the number of attackable arcs within the network.

<table>
<thead>
<tr>
<th>(A, T)</th>
<th>subgame</th>
<th>Large LP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sup. Vec.</td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>Var.</td>
<td>Const.</td>
</tr>
<tr>
<td>(3, 2)</td>
<td>30</td>
<td>0.109</td>
</tr>
<tr>
<td>(3, 3)</td>
<td>211</td>
<td>0.218</td>
</tr>
<tr>
<td>(4, 2)</td>
<td>61</td>
<td>0.297</td>
</tr>
<tr>
<td>(4, 3)</td>
<td>574</td>
<td>0.826</td>
</tr>
</tbody>
</table>

found in the estimation phase. Columns 4 – 5 are the number of variables and the number of constraints in the large LP problems. For the cases where there are only 3 attackable arcs in the network, we restrict arc (3, 7) is not attackable. When the problem is very small, the Large LP approach is faster than our algorithm. However, when the interaction T increases to 3, as in the case of (A, T) = (3, 3), the LP size has 56601 variables and 112987 constraints. The Large LP method solves it with more than 150 seconds, where the subgame decomposition method just needs 0.218 second, about 2 to 3 orders of magnitude faster. In the case of (A, T) = (4, 3), the Large LP method fails to solve within 24 hours while our method requires less than one second.
Table 7.4: The computation versus error in subgame decomposition method.

<table>
<thead>
<tr>
<th>M</th>
<th>((A, T) = (3, 2))</th>
<th>((A, T) = (4, 2))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time vec pt. err</td>
<td>time vec pt. err</td>
</tr>
<tr>
<td>50</td>
<td>0.052 27 2.50% 1.33%</td>
<td>0.151 53 1.75% 0.36%</td>
</tr>
<tr>
<td>100</td>
<td>0.109 30 1.25% 0.21%</td>
<td>0.297 61 1.50% 0.09%</td>
</tr>
<tr>
<td>200</td>
<td>0.202 33 0% 0</td>
<td>0.639 71 0.75% &lt; 10^{-5}</td>
</tr>
</tbody>
</table>

The subgame decomposition method is an approximation method since it estimates the expect payoff functions (or cost-to-go functions for Markov games) with randomly sampled distributions. Statistically speaking, the more it samples, the more accurate estimation it achieves. Table 7.4 compares the computation and the errors of the method for different sampling numbers, which are \(M\) times the number of network states. For example, in the case of \((A, T) = (3, 2)\), the number of network states is \(2^{|A|} = 8\), it randomly samples \(M\) initial distributions to estimate each function. The running times are measured in terms of seconds. “vec” is the number of support vectors that the method obtains. We randomly sample another 400 distributions and compare the estimated values with the true values (obtained in the large LP method). “pt.” is the percentage of these distributions that have relative error (defined as \(1 - \text{est.val./trueval.}\) greater than \(10^{-6}\). “err” is the maximum of these relative errors. No surprising, the running time is linear with \(M\) and the error decreases on \(M\).

Next we increase the number of stages \(T\) and use the subgame decomposition method to solve these problems which can not be solved by the Large LP method due to the problem sizes. The numerical results are shown in Table 7.5. Notice that the large LP method is not available to output the exact equilibrium payoffs, we compare decomposition method with different sampling multipliers \(M\) and take the most accu-
Table 7.5: Decomposition method for larger problems.

<table>
<thead>
<tr>
<th>M</th>
<th>( A = 3 )</th>
<th>( A = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time vec err.avg err.std</td>
<td>time vec err.avg err.std</td>
</tr>
<tr>
<td>( T = 4 )</td>
<td>( \text{LP} = (2.7110^6, 5.4210^6) )</td>
<td>( \text{LP} = (1.3510^8, 2.7010^8) )</td>
</tr>
<tr>
<td>50</td>
<td>0.23 230 0.10% 0.12%</td>
<td>1.33 638 0.44% 0.36%</td>
</tr>
<tr>
<td>100</td>
<td>0.52 412 0.07% 0.09%</td>
<td>3.52 1389 0.31% 0.28%</td>
</tr>
<tr>
<td>200</td>
<td>1.11 629 0.04% 0.06%</td>
<td>11.73 2458 0.23% 0.19%</td>
</tr>
<tr>
<td>400</td>
<td>2.51 1084 N/A N/A</td>
<td>151.69 4469 N/A N/A</td>
</tr>
<tr>
<td>( T = 5 )</td>
<td>( \text{LP} = (1.3010^8, 2.6010^8) )</td>
<td>( \text{LP} = (1.7310^{10}, 3.4610^{10}) )</td>
</tr>
<tr>
<td>50</td>
<td>0.37 235 0.26% 0.11%</td>
<td>2.62 764 1.31% 0.59%</td>
</tr>
<tr>
<td>100</td>
<td>0.98 415 0.09% 0.07%</td>
<td>9.34 1475 1.19% 0.59%</td>
</tr>
<tr>
<td>200</td>
<td>2.45 1042 0.07% 0.07%</td>
<td>111.28 2884 0.34% 0.29%</td>
</tr>
<tr>
<td>400</td>
<td>6.65 1867 N/A N/A</td>
<td>533.12 5913 N/A N/A</td>
</tr>
<tr>
<td>( T = 6 )</td>
<td>( \text{LP} = (6.2510^9, 1.25 \times 10^{10}) )</td>
<td>( \text{LP} = (2.21 \times 10^{12}, 4.4310^{12}) )</td>
</tr>
<tr>
<td>50</td>
<td>0.53 278 0.28% 0.10%</td>
<td>4.32 774 1.49% 0.56%</td>
</tr>
<tr>
<td>100</td>
<td>1.44 564 0.18% 0.09%</td>
<td>40.40 1580 1.22% 0.48%</td>
</tr>
<tr>
<td>200</td>
<td>4.30 1067 0.08% 0.07%</td>
<td>226.04 2894 0.34% 0.28%</td>
</tr>
<tr>
<td>400</td>
<td>43.71 2038 N/A N/A</td>
<td>1104.12 6080 N/A N/A</td>
</tr>
</tbody>
</table>

rate one (i.e. the implementation with the maximum \( M \)) as the proxy of the exact value to estimate the average relative error “err.avg” and its standard deviation “err.std”. “vec” is the number of support vectors found by the method. To give a sense of the size of problems we’re dealing with, Table 7.5 also provide the sizes of equivalent LP problems for the Large LP method with \( \text{LP} = (\text{variables number, constraint number}) \). In all cases, the number of support vectors increases when we raise the number of samples in estimation, meaning that we can better capture the shape of the payoff function. The numbers of support vector can be quite different. For example, in the case of \((A, T) = (3, 4)\), the method can find 278 support vectors with \( M = 50 \), compared with 2038 support vectors found with \( M = 400 \). However, with so many additional support vectors, the accuracy is just improved by about 0.28% in average. At the same time, the computation increases from 0.53 second to 43.71, about two
orders of magnitude. One can make a balance between computation and accuracy based on the computation resource and the solution requirement.

Another interesting observation is that when the number of interaction increases, the error increases as well as the computation. Compare the case of \((A,T) = (4,4), M = 200\) with the case of \((A,T) = (4,6), M = 200\), the error increases from 0.23% to 0.34%. This is because in the subgame decomposition method, the error accumulates from stage to stage, from the lower level subgames to the upper level subgames. The network size (i.e., the number of attackable arcs) or the total number of states can affect the errors too. Compare the case of \((A,T) = (3,5)\) with \(M = 400\) to the case of \((A,T) = (4,5)\) with \(M = 400\), the computation increases from 4.30 seconds to 226.04 seconds and the error increases from 0.08% to 0.34%. Notice the number of network states is doubled from \(2^3 = 8\) to \(2^4 = 16\), the size of LP problem in each subgame is also doubled. Therefore both computation and error increase.
Chapter 8

Conclusions and Future Work

8.1 Conclusions

This dissertation focuses on stochastic network interdiction problems where an attacker selects a set of arcs in a network to interdict with limited resource, with the purpose of minimizing the maximum flows on the resulting network between source(s) and destination(s). The interdiction has random outcome, i.e., the attacked arcs may be completely removed or remain in the network with full capacities under with a given probability. We investigated three classes of such network interdiction problems motivated by specific applications.

The first problem is the one-stage interdiction problem studied in chapter 3. In this model, there is only one round of attack on the network which has random outcomes. The attacker minimizes the expected maximum flows between a source and a destination in the network. Cormican et al. (Cormican et al., 1998) studied this problem before and developed a sequential approximation method (SAA). Borrowing their idea of iteratively increasing the approximation accuracy by considering the average effects of random outcomes, we developed a new solution method (MBB) that integrates the branch and bound techniques with increasingly accurate approximations. Numerical experiments show that MBB is about two orders of magnitude faster than the previous algorithms.

We extended the basic model to incorporate different types of networks, including undirected networks, multi-source/destination networks and networks with uncertain
source/destination. We developed lower and upper bounds separately for each of these extensions and extended our MBB algorithm to solve these problems.

In chapter 5, we extended one-stage interdiction problems to multi-stage interdiction problems where the attacker can interdict the network several times and can adapt sequential strategies based on observing the outcomes of previous attacks. This is a new formulation for multi-stage interdiction problem, which emphasizes the observation of the updated network can affect consequent attacks. We developed a model-predictive approach as in (Castañón and Wohletz, 2009). Instead of solving the exact model, we first solve a lower bound approximation allows for non-integer attack allocations in the second and subsequent waves, but requires integer allocations in the first stage. The optimal solution to the lower bound problem is a binary strategy for the first attack. After carrying out that strategy and observing the result, the attacker has an interdiction problem with one less stage. By doing this iteratively, one can generate binary strategies for all stages. With concrete numerical results, we show that the model-predictive approach often finds the optimal solution. To solve the approximation method, we developed a new set of performance bounds, which are integrated into a branch and bound procedure that extends the single stage approach to multiple stages. Numerical examples show that our method is the best among available methods that can be applied for this problem.

A third class of problems that we studied in this thesis is the interdiction problem where the attacker has only statistical information about the network state, referred as partial information. For instance, the attacker may not observe the result of all attacks, or the attacker may be uncertain about the existence of certain arcs. In these problems, the attacker may gather information from observing the actions of the network. This raises the possibility for the network to deny information to the attacker. We formulated this problem as a zero-sum game with nested information.
To find its equilibrium behavior strategy, we adopted the Linear Programming (LP) formulation developed in (von Stengel, 1996). Notice the size of the LP problem grows exponentially with the number of interactions. To tackle this difficulty, we developed a method, which exploits the nested information structure of the game and decomposes the multi-stage game into a sequence of one-stage subgames. The method estimates the expected payoff of each subgame as a function of its initial conditional probabilities. We showed that the original equilibrium strategies can be found by solving these subgames. For games with Markov structure, we developed an aggregate algorithm that reduces the computation exponentially. Experiments shows that our method is several orders of magnitude faster than the comparable LP approaches of the method in (von Stengel, 1996). More important, for larger problems where the LP approach fails to solve, our method can output high quality solutions within a short time.

### 8.2 Future work

Many questions are interesting for future research topics.

One of the problems of interest is where the attacker's impact on the network can vary. One can allow multiple attack per arc, or stochastic outcomes that are continuous random variables instead of binary random variables. This would happen when an attack may reduce the capacity arbitrarily, to allow for partial success. Under these conditions, the set of future network states resulting from attacks may not be finite. Our lower bound was based on finite partitions of the network state that exploited the binary outcomes. New sets of bounds would be required, along with a branch-and-bound scheme to exploit these bounds.

Another interesting direction is to consider a network with multiple commodities, where the flows are heterogeneous, meaning that each commodity has its own source
and destination, unlike the multi-source/destinations problem in our extension. The attacker is to minimize the total flows of these commodities. One can tackle this problem by separating the flow on each arc into different commodities' flows, the sum of which subjects to the capacity constraint.

We studied a zero-sum game where the attacker has partial information about the network and developed solutions based on the nested information structure of the game. What if the defender also has partial information? It's reasonable to assume that the defender may not completely observe the attacker's actions. This assumption breaks the nested information structure and poses a challenge in developing new efficient algorithms to solve this problem.

Our branch and bound approach developed for stochastic network interdiction problems can be applied to more general mixed-integer stochastic problems. Notice our method just requires a lower bound approximation, which can be increased and eventually equivalent to the original problem by refining the partitions of the space of random outcomes. Therefore our method can be extended to solve any mixed-integer stochastic problems that have the objective function being convex on the random outcomes. It's meaningful to check whether our method can compete with the methods designed specifically for these problems.

The foundation and the premise of our subgame decomposition method is the nested information structure. For games with Markov structure, the computation can be enhanced dramatically. One can apply this method to more general zero-sum games that possess such structures. They can be problems beyond the network interdiction domain. It's interesting to see whether our method can help to solve those problems.
Appendices
Appendix A

The proof of Theorem 6.2.9

**Theorem A.0.1** Let \( \bar{u}, \bar{v} \) be the strategies obtained in procedure (11), then \( \bar{u}, \bar{v} \) are the saddle-point strategies of the original game.

**Proof:** Denote the corresponding part of strategy \( \bar{v} \) in the subgame \( G(I^X, t, \beta^t) \) (not just the first interaction in \( G(I^X, t, \beta^t) \)) as \( \bar{v}(I^X,t) \), similarly denote the corresponding part of strategy \( \bar{u} \) in the subgame \( G(I^X, t, \beta^t) \) as \( \bar{u}(I^X,t) \).

Fix player \( Y \)'s strategy \( \bar{v} \) and allow player \( X \) change its strategy \( u \). According to (6.30) where

\[
\beta_{[\sigma^{t-1},a^t,d^t,s^t]}^{I^X,t+1} = \beta_{[\sigma^{t-1},a^t,d^t,s^t]}^{I^X,t+1} / \sum_{\delta^{t-1},s^t} \beta_{[\sigma^{t-1},a^t,d^t,s^t]}^{I^X,t+1}
\]

the conditional distribution \( \beta^{I^X,t+1} \) for subgame \( G(I^X,t+1, \beta^{I^X,t+1}) \) is NOT affected by player \( X \)'s strategy \( u \). Therefore, when \( \bar{v} \) is fixed, the conditional distribution \( \beta^{I^X,t+1} \) on every information set \( I^{X,t+1} \) are fixed.

At the bottom level subgames \( G(I^X,t, \beta^{I^X,t}) \), by theorem 6.2.3, \( \bar{u}(I^X,t) \) and \( \bar{v}(I^X,t) \) are the saddle point strategies with the value \( V(\beta^{I^X,t}) \). For a subgame \( G(I^X,t, \beta^{I^X,t}) \), assume we have shown that \( \bar{u}(I^X,t+1) \) is the best response to \( \bar{v} \) in the lower level subgame \( G(I^X,t+1, \beta^{I^X,t+1}) \) for all \( I^X,t+1 \succ I^X,t \), meaning Player \( X \) cannot get better payoff in subgame \( G(I^X,t+1, \beta^{I^X,t+1}) \) by changing \( \bar{u}(I^X,t+1) \) for any
According to problem (6.18) where

$$\max_{x_t \in X_t} \min_{y_t \in Y_t} \sum_{I_{X,t+1} \supset I_{X,t}} P_{I_{X,t+1}}(x_t, y_t) V_{I_{X,t+1}}(\beta^{I_{X,t+1}}(x_t, y_t)),$$

the expected payoff in the subgame $G(I_{X,t}, \beta^{I_{X,t}})$ is the weighted sum of the expected payoffs of its lower level subgames $G(I_{X,t+1}, \beta^{I_{X,t+1}})$ with the weight is calculated as

$$P_{I_{X,t+1}}(x_t, y_t) := x_t \sum_{d_t \in I_{X,t+1}} \sum_{(s_t, a_t) \in \gamma_{I_{X,t},d_t}} y_{t+1}(\sigma_t, a_t) R_{I_{X,t}}(\sigma_t, s_t).$$

By theorem 6.2.3, $u_{I_{X,t},a_t}$ (or equivalently $x_t$) is the saddle point strategy for the one stage game, therefore it optimizes the weight of $P_{I_{X,t+1}}$ for all $I_{X,t+1} \supset I_{X,t}$. As a result, player X can not get better payoff by changing $u(I_{X,t+1})$ for all $I_{X,t+1} \supset I_{X,t}$. Therefore, $u(I_{X,t})$ is the best response to $\hat{v}$ in the subgame $G(I_{X,t}, \beta^{I_{X,t}})$. By induction from $t = T, \ldots, 1$, we show that $u$ is the best response to $\hat{v}$.

Next we will show that $\hat{v}$ is the best response to $u$. According to procedure (11), for any subgame $G(I_{X,t}, \beta^{I_{X,t}})$, $\hat{v}_{I_{X,t},d_t}$ is the (part of) optimal solution of the corresponding problem (6.23), which is converted from the min-max problem in (6.20)

$$\min_{v \in V} \max_{u \in U} J_{I_{X,t}}(u, v),$$

without changing $v$. Here $J_{I_{X,t}}(u, v)$ is

$$J_{I_{X,t}}(u, v) = \sum_{\sigma_{t-1} \in I_{X,t}} \beta^{I_{X,t}} \sum_{a_t} \sum_{d_t} u(I_{X,t}, a_t, d_t, a_t) \sum_{\sigma_t \in \Lambda_{I_{X,t+1}}} \lambda^{I_{X,t+1}} \sum_{k=1} K^{I_{X,t+1}} \sum_{s_t \in S_{t-1}} \lambda^{I_{X,t+1}} \sum_{s'} \gamma_{s_t, s'} Q_k(s_{t-1}, s', a_t, d_t).$$

Notice in procedure (11), $u_{I_{X,t},d_t}$ and $\lambda^{I_{X,t+1}}$ in the one stage problem $G(I_{X,t}, \beta^{I_{X,t}})$ are derived from $w^{I_{X,t}}$. When player X's strategy $w^{I_{X,t}}$ for all $I_{X,t}$ is fixed, $u_{I_{X,t},d_t}$
and $\lambda^{I_{X,t+1}}$ are also fixed. (note that if $w^{*I_{X,t}}$ is not obtained via (6.35) in procedure (11), $\lambda^{I_{X,t+1}}$ is not necessary fixed, as shown in the last paragraph of subsection 6.2.4.) By theorem 6.2.3, $\bar{v}_{I_{Y,t},d^t}$ is the saddle point strategy in the one stage subgame $G(I_{X,t}, \beta^{I_{X,t}})$. It must satisfy that, for any $I_{Y,t} = [\sigma^{t-1}, a^t] \succ I_{X,t}$

$$
\sum_{d^t} \bar{v}_{[\sigma^{t-1}, a^t], d^t} \max_{\lambda^{I_{X,t+1}} \in \Lambda_{I_{X,t+1}}} \sum_{k=1}^{K_{I_{X,t+1}}} \lambda_k^{I_{X,t+1}} \sum_{s^t} t[\sigma^{t-1}, a^t, d^t, s^t, q_k[\sigma^{t-1}, a^t, d^t, s^t]]
$$

Note that this minimization does not depend on $\beta^{I_{X,t}}$, which means even when $\beta^{I_{X,t}}$ is changed by player Y's past strategies on the subgames in the upper levels, the optimality of $\bar{v}_{I_{Y,t},d^t}$ at $I_{Y,t}$ still holds. For player Y, the only possible way to get better payoffs at $I_{Y,t}$ is to change its future strategies after $I_{X,t+1}$ to make the expected payoffs of nodes in the lower level information sets $I_{X,t+1}$ become better and then adjust $v_{I_{Y,t},d^t}$ based on these payoffs. Now we're going to show by changing its future strategies after $I_{X,t+1}$ alone, player Y cannot improve the payoffs on nodes in $I_{X,t+1}$.

When $t = T$, this is trivial since nodes in $I_{X,T+1}$ are leaf nodes, where payoffs are given. Now we just need to show player Y can not get better payoffs for nodes in $I_{X,t}$ by changing its strategy after $I_{X,t}$. Let $q_{I_{X,t}}, q_{T-1}(w)$ defined in (6.37) and (6.36), respectively. That is

$$
q_{I_{X,t}} := \sum_{k=1}^{K_{I_{X,t}}} \lambda_k^{I_{X,t}} I_{X,t,k}.
$$
For any $\sigma^{t-1}$ in $I^{X,t}$,

$$
q(w)_{\sigma^{t-1}} := \sum_{\alpha} \min_v \sum_{d^t} v[\sigma^{t-1},\alpha_t,d_t] dt \sum_{k=1}^{K^{I^{X,t+1}}} w^X_{t+1} \sum_{s^t} T[\sigma^{t-1},\alpha_t,d_t] s^t q_k[\sigma^{t-1},\alpha_t,d_t,s^t].
$$

By these definitions, for nodes in $I^{X,t}$, $q_{I^{X,t}}$ are their expected payoffs in the one-stage subgame $G(I^{X,t-1},\beta^{I^{X,t-1}})$ given the saddle point strategy $w^{*I^{X,t-1}}$ and $q_{I^{X,t}}(w^{*I^{X,t}})$ are their expected payoffs in the one-stage subgame $G(I^{X,t},\beta^{I^{X,t}})$ given the saddle point strategy $w^{*I^{X,t}}$. The property $P1$ in theorem 6.2.8 states

$$
q_{I^{X,t}}(w^{*I^{X,t}}) = q_{I^{X,t},\sigma^{t-1}} \text{ if } \beta_{t-1}^{I^{X,t}} > 0,
$$

which means for each node $\sigma^{t-1}$ with positive realization probabilities (i.e., $\beta_{t-1}^{I^{X,t}} > 0$), its expected payoffs in the subgames $G(I^{X,t-1},\beta_{t-1}^{I^{X,t-1}})$ and $G(I^{X,t},\beta^{I^{X,t}})$ are the same. The property $P1$ in theorem 6.2.8 also states

$$
q_{I^{X,t}}(w^{*I^{X,t}}) \geq q_{I^{X,t},\sigma^{t-1}},
$$

meaning that for each node $\sigma^{t-1}$, its expected value in the lower level subgame $G(I^{X,t},\beta^{I^{X,t}})$ is the upper bound of its expected value in the upper level subgame $G(I^{X,t-1},\beta_{t-1}^{I^{X,t-1}})$. Therefore, player $Y$ will not choose $\sigma^{t-1}$ based on $q_{I^{X,t}}(w^{*I^{X,t}})$ if it does not choose $\sigma^{t-1}$ based on $q_{I^{X,t},\sigma^{t-1}}$. As a result, player $Y$ can not get better payoffs on nodes in $I^{X,t}$ by changing its strategies after $I^{X,t}$ alone. By induction from $t = T, \ldots, 1$, we show that $\hat{v}$ is the best response to $\hat{u}$.

Because $\hat{u}, \hat{v}$ are the best response to each other in the multi-stage game, $\hat{u}, \hat{v}$ are the saddle point strategies.
References


Jiefu Zheng received his B.S in Computational Mathematics in June 2004 and M.S. in Operations Research in June 2007, Nanjing University, Nanjing, China. In September 2007, he began to pursue a PhD in Systems Engineering at Boston University. Starting on September, 2010, he focused on network interdiction problems under the guidance of Professor David Castañón. Besides his research, he is also interested in optimization, machine learning and data analysis. Jiefu Zheng has been married with Qi Wang in December, 2007. They have a lovely daughter Alice, who was born in August, 2010.