Equilibrium Strategies for Multiple Interdictors on a Common Network

Harikrishnan Sreekumaran∗ Ashish R. Hota† Andrew L. Liu‡ Nelson A. Uhan§ Shreyas Sundaram¶

June 26, 2017

Abstract

In this work, we introduce multi-interdictor games, which model interactions among multiple interdictors with differing objectives operating on a common network. As a starting point, we focus on shortest path multi-interdictor (SPMI) games, where multiple interdictors try to increase the shortest path lengths of their own adversaries attempting to traverse a common network. We first establish results regarding the existence of equilibria for SPMI games under both discrete and continuous interdiction strategies. To compute such an equilibrium, we present a reformulation of the SPMI game, which leads to a generalized Nash equilibrium problem (GNEP) with non-shared constraints. While such a problem is computationally challenging in general, we show that under continuous interdiction actions, a SPMI game can be formulated as a linear complementarity problem and solved by Lemke’s algorithm. In addition, we present decentralized heuristic algorithms based on best response dynamics for games under both continuous and discrete interdiction strategies. Finally, we establish theoretical lower bounds on the worst-case efficiency loss of equilibria in SPMI games, with such loss caused by the lack of coordination among noncooperative interdictors, and use the decentralized algorithms to numerically study the average-case efficiency loss.

1 Introduction

In an interdiction problem, an agent attempts to limit the actions of an adversary operating on a network by intentionally disrupting certain components of the network. Such problems are usually modeled in the framework of leader-follower games and can be formulated as bilevel optimization problems. Interdiction models have been used in various military and homeland security applications, such as dismantling drug traffic networks [51], preventing nuclear smuggling [34] and planning tactical air strikes [24]. Interdiction models have also found applications in other areas such as controlling the spread of pandemics [4] and defending attacks on computer communication networks [44].

Traditionally, interdiction problems have been analyzed from a centralized perspective; namely, a single agent is assumed to analyze, compute and implement interdiction strategies. In many situations, however, it might be desirable and even necessary to consider an interdiction problem from a decentralized perspective. Arguably the most prominent example of such situations today is the war against the terrorist group, the

∗American Airlines, Fort Worth, Texas USA. Email: harikrishnan.sreekumaran@gmail.com
†School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA. Email: ahota@purdue.edu
‡Corresponding author. School of Industrial Engineering, Purdue University, West Lafayette, Indiana, USA. Email: andrewliu@purdue.edu
§Mathematics Department, United States Naval Academy, Annapolis, Maryland, USA. Email: uhan@usna.edu
¶School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA. Email: sundara2@purdue.edu
Islamic State of Iraq and Syria (ISIS, also known as ISIL or Daesh). It is believed that oil smuggling is the “biggest single source of revenue” of ISIS [45], and hence, a sensible strategy to defeat ISIS is to disrupt their oil smuggling. Such a strategy indeed has been deployed by the multiple parties involved in the war [1,42]. The parties involved, however, which include the US-led coalition, Russia, Turkey, Iran, among others, do not operate as a single coalition, and often do not share information [26]. Without any coordination between the parties, one might expect that a decentralized interdiction strategy may be inefficient compared to one determined by a central decision maker. A central decision maker in the war against ISIS is of course impractical, and hence, we would like to understand better the equilibrium state of such settings with multiple interdictors on a common network, and especially the efficiency loss due to the lack of cooperation among the interdictors. This is both the motivation and the focus of this paper.

In this paper, we introduce decentralized multiple interdictor games, in which multiple agents with differing objectives are interested in interdicting parts of a common network. We focus on a specific class of these games, which we call shortest path multi-interdictor (SPMI) games. We investigate various properties of equilibria in SPMI games, including their existence and uniqueness, and propose algorithms to compute equilibria of these games. Using these algorithms, we also conduct numerical studies on the efficiency loss of equilibria in the SPMI game compared to optimal solutions obtained through centralized decision making.

Decentralized network interdiction games, as will be formally defined in Section 2, appear to be new. To the best of our knowledge, there has been no previous research on such games. As a result, not much is known about the inefficiency of equilibria for these games or intervention strategies to reduce such inefficiencies. There has been a considerable amount of work, however, on interdiction problems from a centralized decision-maker’s perspective. As mentioned earlier, interdiction problems have been studied in the context of various military and security applications. For extensive reviews of the existing academic literature on interdiction problems, we refer the readers to Church et al. [9] and Smith and Lim [44].

There have also been many studies on the inefficiency of equilibria in other game-theoretic settings. Most of the efforts have been focused on routing games [37,50], in which selfish agents route traffic through a congested network, and congestion games [40], a generalization of routing games. Some examples include [5,8,10,12,41,46]. Several researchers have also studied the inefficiency of equilibria in network formation games, in which agents form a network subject to potentially conflicting connectivity goals [2,3,17]. The inefficiency of equilibria has been studied in other games as well, such as facility location games [49], scheduling games [31], and resource allocation games [28,29]. Almost all of the work described above study the worst-case inefficiency of a given equilibrium concept. Although a few researchers have studied the average inefficiency of equilibria, either theoretically or empirically, and have used it as a basis to design interventions to reduce the inefficiency of equilibria [11,47], research in this direction has not received much attention.

One potential reason for the lack of attention paid to multiple interdictor games may be that such games often involve nondifferentiability, as each interdictor’s optimization problem usually entails a max-min type of objective function. Games involving nondifferentiable functions are generally challenging, in terms of both theoretical analysis of their equilibria and computing an equilibrium. While in some cases (such as in the case of shortest path interdiction), a smooth formulation (through total unimodularity and duality) is possible, such a reformulation will yield a generalized Nash equilibrium problem (GNEP), in which both the agents’ objective functions as well as their feasible action spaces depend on other agents’ actions. Although the conceptual framework of GNEPs can be dated to Debreu [14], rigorous theoretical and algorithmic treatments of GNEPs began much later (see [19], for example, for a literature review). Several techniques have been proposed to solve GNEPs, including penalty-based approaches [20,23], variational-inequality-based approaches [36], Newton’s method [15], projection methods [52], and relaxation approaches [32,48]. Most of the work on GNEPs has focused on games with shared constraints due to their tractability [18,25]. In such games, a set of identical constraints appear in each agent’s feasible action set. However, as will be seen later, in a typical decentralized network interdiction game, the constraints involving multiple agents’
actions that appear in each agent’s action space are not identical. As a result, such games give rise to more challenging instances of GNEPs.

Based on the discussions above, the major contributions of this work are as follows.

• First, we establish the existence of pure-strategy equilibria for SPMI games with continuous interdiction. In SPMI games with discrete interdiction, the existence of a pure strategy Nash equilibrium (PNE) is more subtle. We first demonstrate that a PNE does not necessarily exist in SPMI games with discrete interdiction. However, when all agents have the same source-target pairs (i.e., multiple agents try to achieve a common goal independently), a PNE exists in these games.

• Second, for SPMI games under continuous interdiction, we show that each agent’s optimization problem can be reformulated as a linear programming problem. As a result, the equilibrium conditions of the game can be reformulated as a linear complementarity problem with some favorable properties, allowing it to be solved by the well-known Lemke algorithm [33]. For SPMI games with discrete interdiction (and continuous interdiction), we present decentralized algorithms for finding an equilibrium, based on the well-known best-response dynamics (or Gauss-Seidel iterative) approach. While such an approach is only a heuristic method in general, convergence can be established for the special case when the agents have common source-target pairs. For more general cases, we obtain encouraging numerical results for the performance of the method on several classes of network structures.

• Third, we measure the efficiency loss in SPMI games due to the lack of coordination among noncooperative interdictors, compared to a centralized interdiction strategy (that is, a strategy implemented by a single interdictor with respect to all the adversaries). In particular, we establish a theoretical lower bound for the worst-case price of anarchy of SPMI games under continuous interdiction. Such an efficiency loss measure, however, may be too conservative, and so we use the decentralized algorithms to numerically quantify the average-case efficiency loss over some instances of SPMI games. These results can help central authorities design mechanisms to reduce such efficiency losses for practical instances.

The remainder of this paper is organized as follows. We begin in Section 2 with definitions and formulations of general multi-interdictor games and the specific SPMI games. In Section 3, we present the main theoretical results of the paper, including an analysis of the existence and uniqueness of equilibria in SPMI games. In Section 4 we investigate algorithms for solving SPMI games. We describe a centralized algorithm based on a linear complementarity formulation, as well as decentralized algorithms for computing equilibria of SPMI games. We also present the results of our computational experiments with these algorithms for computing equilibria as well as quantifying the price of anarchy for various instances. Finally, in Section 5, we provide some concluding remarks.

2 Games with Multiple Interdictors on a Common Network

2.1 General Formulation

Network interdiction problems involve interactions between two types of parties – adversaries and interdictors – with conflicting interests. An adversary operates on a network and attempts to optimize some objective, such as the flow between two nodes. An interdictor tries to limit an adversary’s objective by changing elements of the network, such as the arc capacities. Such interactions have historically been viewed from a leader-follower-game perspective. The interdictor acts as the leader and chooses an action while anticipating the adversary’s potential responses, while the adversary acts as the follower and makes a move after observing the interdictor’s actions. From the interdictor’s perspective, this captures the pessimistic viewpoint of guarding against the worst possible result given its actions.
In this work, we consider strategic interactions among multiple interdictors who operate on a common network. The interdictors may each have their own adversary or they may have a common adversary. If there are multiple adversaries, we assume there is no strategic interaction among them. We also assume that the interdictors are allies in the sense that they are not interested in deliberately impeding each other.

Formally, we have a set $\mathcal{F} = \{1, \ldots, F\}$ of interdictors or agents, who operate on a network $G = (V, A)$, where $V$ is the set of nodes and $A$ is the set of arcs. Each agent’s actions or decisions correspond to interdicting each arc of the network with varying intensity: the decision variables of agent $f \in \mathcal{F}$ are denoted by $x^f \in X^f \subset \mathbb{R}^{|A|}$, where $X^f$ is an abstract set that constrains agent $f$’s decisions. For any agent $f \in \mathcal{F}$, let $x^{-f}$ denote the collection of all the other agents’ decision variables; that is, $x^{-f} = (x^1, \ldots, x^{f-1}, x^{f+1}, \ldots, x^F)$. The network obtained after every agent executes its decisions or interdiction strategies is called the aftermath network. The strategic interaction between the agents occurs due to the fact that the properties of each arc in the aftermath network are affected by the combined decisions of all the agents.

In addition to the abstract constraint set $X^f$, we assume that each agent $f \in \mathcal{F}$ has a total interdiction budget of $b^f$. The cost of interdicting an arc is linear in the intensity of interdiction; in particular, agent $f$’s cost of interdicting arc $(u, v)$ by $x^f_{uv}$ units is $c^f_{uv} x^f_{uv}$. Without loss of generality, we assume that $b^f > 0$ and $c^f_{uv} > 0$ for each arc $(u, v) \in A$ and for each agent $f \in \mathcal{F}$. The optimization problem for each agent $f \in \mathcal{F}$ is:

$$\begin{align*}
\text{maximize} & \quad \theta_f(x^f, x^{-f}) \\
\text{subject to} & \quad \sum_{(u,v) \in A} c^f_{uv} x^f_{uv} \leq b^f, \\
& \quad x^f \in X^f,
\end{align*}$$

where the objective function $\theta_f$ is agent $f$’s obstruction function, or measure of how much agent $f$’s adversary has been obstructed. Henceforth, we refer to the game in which each agent $f \in \mathcal{F}$ solves the above optimization problem $[1]$ as a decentralized multi-interdictor game. As a starting point, we restrict our attention to simultaneous-move games with complete information. Simultaneous-move means that the agents must make their decisions without being aware of the other agents’ decisions. A complete information game means that the number of agents, their payoffs and their feasible action spaces are common knowledge to all the agents.

The obstruction function $\theta_f$ can capture various types of interdiction problems. Typically $\theta_f$ is the (implicit) optimal value function of the adversary’s network optimization problem parametrized by the agents’ decisions. For example, $\theta_f$ might be the minimum flow cost or path length subject to flow conservation, arc capacity and side constraints in the aftermath network.

Suppose that a central planner, with a comprehensive view of the network and the agents’ objectives, could pool the agents’ interdiction resources and determine an interdiction strategy that maximizes some global measure of how much the agents’ adversaries have been obstructed. Let $\theta^c(x^1, \ldots, x^F)$ represent the global obstruction function for a given interdiction strategy $(x^1, \ldots, x^F)$. The central planner’s problem corresponding to the multi-interdictor game $[1]$ is then:

$$\begin{align*}
\text{maximize} & \quad \theta^c(x^1, \ldots, x^F) \\
\text{subject to} & \quad \sum_{f \in \mathcal{F}} \sum_{(u,v) \in A} c^f_{uv} x^f_{uv} \leq \sum_{f \in \mathcal{F}} b^f, \\
& \quad x^f \in X^f \quad \forall f \in \mathcal{F}.
\end{align*}$$

Without loss of generality, we assume that $\theta^c(\cdot) \geq 0$ for any feasible $(x^1, \ldots, x^F)$. We refer to (2) as the
centralized problem, and focus primarily on when the global obstruction function is utilitarian; that is,

\[ \theta^c(x^1, \ldots, x^F) := \sum_{f \in F} \theta^f(x^f, x^{-f}). \]

Note also that we assume the resources involved in the budgetary constraints may be “pooled” amongst the interdictors. Such resources may then be allocated optimally by the central planner. The case where the resources are not shareable can easily be modeled by enforcing each interdictor’s resource constraints separately in the central planner’s problem.

As mentioned earlier, one of the goals of this work is to quantify the inefficiency of an equilibrium of a decentralized multi-interdictor game – a decentralized solution to problem (1) – relative to a centrally planned optimal solution – an optimal solution to problem (2). A commonly used measure of such inefficiency is the \textit{price of anarchy}. Formally speaking, let \( N_I \) be the set of all equilibria corresponding to a specific instance \( I \). (In the context of multi-interdictor games, an instance consists of the network, obstruction functions, interdiction budgets, and costs.) For the same instance \( I \), let \( (x^1^*, \ldots, x^F^*) \) denote a globally optimal solution to the centralized problem (2). Then the price of anarchy of the instance \( I \) is defined as

\[
p(I) := \max_{(x^1_N, \ldots, x^F_N) \in N_I} \frac{\theta^c(x^1^*, \ldots, x^F^*)}{\theta^c(x^1_N, \ldots, x^F_N)}.
\]

(3)

Let \( I \) be the set of all instances of a game. We assume implicitly that for all \( I \in I \), the set \( N_I \) is nonempty and a globally optimal solution to the centralized problem exists. By convention, \( p \) is set to 1 if the worst equilibrium as well as the globally optimal solution to the centralized problem both have zero objective value. If the worst equilibrium has a zero objective value while the globally optimal value of the centralized problem is positive, \( p \) is set to be infinity. In addition to the price of anarchy for an instance of a game, we also define the worst-case price of anarchy over all instances of the game (denoted as \( w.p.o.a \)) as follows:

\[
w.p.o.a := \sup_{I \in I} p(I).
\]

(4)

We wish to study the efficiency loss of the class of multi-interdictor games. The worst-case price of anarchy provides a way to measure this. However, there are two major difficulties associated with this as an efficiency measure. First, it is well-known that the worst-case price of anarchy can be a very conservative measure of efficiency loss, since the worst case may only happen with pathological instances. Second, explicit theoretical bounds on the worst-case price of anarchy may be difficult to obtain for general classes of games. In fact most of the related research has focused on identifying classes of games for which such bounds may be derived. In this work, we show how our proposed decentralized algorithms can be used to numerically study the \textit{average-case efficiency loss} (denoted by \( a.e.l \)). Let \( I' \subset I \) denote a finite subset of instances, and let \( |I'| \) denote the cardinality of the set \( I' \). Then

\[
a.e.l(I') := \frac{1}{|I'|} \sum_{I \in I'} p(I).
\]

(5)

In other words, the average-case efficiency loss is the average value of \( p(I) \) as defined in (3) over a set of sampled instances \( I' \subset I \) of a game.

As mentioned above, the generic form of problem (1) can be used to describe various network interdiction settings. To start with models that are both theoretically and computationally tractable, we focus on shortest-path multi-interdictor games, which we describe in detail next.

\section{2.2 Shortest Path Multi-interdictor Games}

As the name suggests, \textit{shortest path multi-interdictor (SPMI)} games involve agents or interdictors whose adversaries are interested in the shortest path between source-target node pairs on a network. Interdictors
act in advance to increase the length of the shortest path of their respective adversaries by interdicting (in particular, lengthening) arcs on the network.

To describe these games formally, we build upon the setup of the general multi-interdictor game described in Section 2.1. Each agent \( f \in \mathcal{F} \) has a target node \( t^f \in V \) that it wishes to protect from an adversary at source node \( s^f \in V \) by maximizing the length of the shortest path between the two nodes. The agents achieve this goal by committing some resources (e.g. monetary spending) to increase the individual paths available to their respective adversaries by interdicting (in particular, lengthening) arcs on the network: the decision variable \( x^f_{uv} \) represents the contribution of agent \( f \in \mathcal{F} \) towards lengthening arc \((u,v) \in A\). The arc length \( d_{uv}(x^f, x^{-f}) \) of arc \((u,v) \in A\) in the aftermath network depends on the decisions of all the agents.

We consider two types of interdiction. The first type of interdiction is \textit{continuous}: in particular,

\[
X^f := \{ x^f \in \mathbb{R}^{|A|} : x^f_{uv} \geq 0 \quad \forall (u,v) \in A \}.
\]

The arc lengths after an interdiction strategy \((x^1, \ldots, x^F)\) has been executed are

\[
d_{uv}(x^1, \ldots, x^F) = d^0_{uv} + \sum_{f \in \mathcal{F}} x^f_{uv} \quad \forall (u,v) \in A,
\]

where \( x^f_{uv} \) captures how much agent \( f \) extends the length of arc \((u,v)\). We assume that \( d^0_{uv} \geq 0 \) for all \((u,v) \in A\). The second type of interdiction is \textit{discrete}: in this case,

\[
X^f := \{ x^f \in \mathbb{R}^{|A|} : x^f_{uv} \in \{0, 1\} \quad \forall (u,v) \in A \}
\]

and the arc lengths in the aftermath network are

\[
d_{uv}(x^1, \ldots, x^F) = d^0_{uv} + e_{uv} \max_{f \in \mathcal{F}} x^f_{uv} \quad \forall (u,v) \in A,
\]

where \( e_{uv} \in \mathbb{R}_{\geq 0} \) is the fixed extension of arc \((u,v)\). In other words, the length of an arc is extended by a fixed amount if at least one agent decides to interdict it.

Let \( P^f = \{ p_1^f, p_2^f, \ldots, p_k^f \} \) be the set of \( s^f \) to \( t^f \) paths available to agent \( f \in \mathcal{F} \). The length of a path \( p \in P^f \) is given by

\[
d_p(x^1, \ldots, x^F) = \sum_{(u,v) \in p} d_{uv}(x^1, \ldots, x^F),
\]

where \( d_{uv}(x^1, \ldots, x^F) \) is as defined in equation (6) for continuous interdiction, and as defined in (7) for the discrete case. The optimization problem for each interdicting agent \( f \in \mathcal{F} \) is then:

\[
\begin{align*}
\text{maximize} & \quad \theta^f(x^f, x^{-f}) = \min_{p \in P^f} d_p(x^f, x^{-f}) \\
\text{subject to} & \quad \sum_{(u,v) \in A} c^f_{uv} x^f_{uv} \leq b^f, \quad x^f \in X^f.
\end{align*}
\]

Under continuous interdiction and assuming that \( X^f \) is nonempty, convex and compact, the feasible strategy set for agent \( f \), given by \( \{ x^f \in X^f \mid \sum_{(u,v) \in A} c^f_{uv} x^f_{uv} \leq b^f \} \) is also convex and compact. To rule out uninteresting cases, we also assume that the feasible set for each agent is also nonempty (meaning that each agent has the budget to interdict some arcs). Moreover, given an \( x^{-f} \), the objective function in (9) is the minimum of a set of affine functions of \( x^f \), and therefore continuous in \( x^f \). Thus, by Weierstrass’s extreme value theorem, each agent has an optimal strategy given the strategies of the other agents. Note, however, that the objective function in (9) is not differentiable with respect to \( x^f \) in general.

For SPMI games with discrete interdiction, the feasible strategy set for each agent is finite. Therefore an optimal solution to each agent’s problem always exists with a given \( x^{-f} \). In the following section, we analyze the existence and uniqueness of pure strategy Nash Equilibria for SPMI games, under both continuous and discrete settings.
3 Game Structure and Analysis

3.1 Existence of Equilibria

We first consider the existence of a Nash equilibrium of a SPMI game when interdiction decisions are continuous. The key is to show that the objective function in (9) \( \theta^f(x^f, x^{-f}) \) is concave in \( x^f \), despite the fact that it is not differentiable.

**Proposition 1.** Assume that the abstract set \( X_f \) in (9) is nonempty, convex and compact for each \( f \in F \). The SPMI game under continuous interdiction – each agent \( f \in F \) solves the problem (9) with \( d_p(x^f, x^{-f}) \) defined as in (8) and (6) – has a pure strategy Nash equilibrium.

**Proof.** Based on the assumption, the feasible region in (9) is nonempty, convex and compact. With a fixed \( x^{-f} \), the objective function of agent \( f \) is the minimum of a finite set of affine functions in \( x^f \), and therefore, is concave with respect to \( x^f \) (Cf. [7]). Consequently, the SPMI game belongs to the class of “concave games,” introduced in Rosen [39], and it is shown in [39] that a pure-strategy Nash equilibrium always exists for a concave game.

Under discrete interdiction, the existence of a PNE is not always guaranteed when different interdictors are competing against different adversaries. We illustrate the nonexistence of PNE in Example 1 below.

**Example 1.** Consider the network given in Figure 1.

![Figure 1: Network topology for the SPMI game in Example 1](image)

In this game, there are two agents – agent 1 and agent 2 – who are attempting to maximize the lengths of the \( s^1-t^1 \) paths and \( s^2-t^2 \) paths respectively. Note that \( t^1 = t^2 \). The data for the problem, including initial arc lengths, cost of interdiction and arc extensions are given below in Table 1.

<table>
<thead>
<tr>
<th>Arc tag</th>
<th>Initial length</th>
<th>Arc extension</th>
<th>Cost to player 1</th>
<th>Cost to player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>7</td>
<td>0.5</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>20</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>1.5</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>6</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>1</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>f</td>
<td>1</td>
<td>6</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 1: Network data for Example 1

Suppose that the budgets are \( b^1 = 8 \) and \( b^2 = 15 \). As a result, player 1 can either interdict the arcs \( a, b \) and \( c \) one at a time, or the arcs \( a \) and \( c \) simultaneously. Similarly, player 2 can either interdict arc \( d \) or arc \( f \).
Thus, player 1 has four feasible pure strategies and player 2 has two feasible pure strategies. The strategy
tuples along with the corresponding payoffs for each player are summarized in Table 2. It is easy to verify
that for any joint strategy profile, there is a player who would prefer to deviate unilaterally. Therefore, this
instance of the SPMI game does not possess a PNE.

<table>
<thead>
<tr>
<th>Player 1 / Player 2</th>
<th>d</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>6, 1</td>
<td>0, 0</td>
</tr>
<tr>
<td>c</td>
<td>7, 1</td>
<td>1.5, 1.5</td>
</tr>
<tr>
<td>(a, c)</td>
<td>7.5, 1</td>
<td>1.5, 1.5</td>
</tr>
<tr>
<td>b</td>
<td>7, 1</td>
<td>2, 0</td>
</tr>
</tbody>
</table>

Table 2: Payoff combinations for Example 1

In the previous example, the agents have a common target node, but different source nodes. However,
in the class of games in which the interdictors have a common adversary, i.e., when each agent maximizes
the shortest path between a common source-target pair, we can show that SPMI games under discrete inter-
diction possess a PNE.

Consider the SPMI game where each agent is trying to maximize the shortest path lengths between
nodes s and t. Since the objective function of each agent is the same, we can write the following centralized
optimization problem to maximize the shortest s-t path distance subject to the individual agents’ budget
constraints. Let $P_{st}$ be the set of s-t paths in the network. The centralized optimization problem is:

$$\text{maximize} \quad \min_{p \in P_{st}} d_p(x^1, x^2, \ldots, x^F)$$

subject to

$$\sum_{(u,v) \in A} c_{uv}^f x_{uv}^f \leq b^f \quad \forall f \in F,$$

$$x_{uv}^f \in \{0, 1\} \quad \forall (u, v) \in A, f \in F. \tag{10}$$

The feasible solution space of the above problem is finite under individual agents’ budget constraints. There-
fore, the centralized problem always has a maximum. Furthermore, the optimal solution to this problem is a
PNE of the SPMI game as we show in the following result.

**Proposition 2.** Suppose the source and target for each agent in a SPMI game under discrete interdiction,
are the same. Let $x^*$ denote the optimal solution of the centralized problem $\text{(10)}$. Then $x^*$ is a PNE to the
SPMI game.

**Proof.** Assume the contrary, and suppose that there is an agent $h$ for whom there exists a unilateral deviation
$x^f$ that strictly increases the path distance from source to target. By assumption, $x^h$ is feasible for the
budgetary constraints for agent $h$. Therefore, $\bar{x} \equiv (x^h, x^{*-h})$ is feasible for $\text{(10)}$ with a strictly larger
objective value. Clearly this is a contradiction to the optimality of $x^*$ for $\text{(10)}$.

### 3.2 Non-Uniqueness of Equilibria

Establishing sufficient conditions for a SPMI game to have a unique equilibrium is quite difficult. However,
it is easy to find simple instances of SPMI games for which multiple equilibria exist. We give two such
examples below.

**Example 2.** Consider the following instance, based on the network in Figure 2. There are 2 agents: agent 1
has an adversary with source node 1 and target node 5; agent 2 has an adversary with source node 1 and
target node 6. The initial arc lengths are 0, interdiction is continuous, and the interdiction costs are the same.
for both agents and are given in the arc labels in Figure 2. Both agents have a budget of 1. Consider the case when $\epsilon = 2$. In this case, it is straightforward to see that the source-target path lengths for each agent must be equal at an equilibrium: if the path lengths are unequal, an agent could improve its objective function by equalizing the path lengths. Therefore, in this example, any combination of decision variables that results in a shortest path length of $2/3$ for each agent is a Nash equilibrium, and there is a continuum of such decision variable combinations. Indeed, some of such equilibria are given in Table 3 in Section 5.

Example 3. Under discrete interdiction on the same underlying network, an interesting situation occurs when $\epsilon = 0$, both agents have a budget of 1, and the arc extensions are all set to 1. In this case, an equilibrium occurs when the arcs $(1, 4)$ and $(1, 2)$ are interdicted by one agent each. However, there exist equilibria that have inferior objective values for both agents. Indeed, the extreme case of neither agent interdicting any arc can easily seen to be an equilibrium. This point in fact is a social utility minimizer over the set of feasible action combinations for the two agents.

4 Computing a Nash Equilibrium

In this section we discuss algorithms to compute equilibria of SPMI games. While the general formulation with each agent solving (9) is sufficient for showing existence of equilibria, such a formulation is not amenable for computing an equilibrium mainly due to the ‘min’ function in the objective function. In this section, using a well-known reformulation of shortest path problems (through total unimodularity and linear programming duality), we formulate the SPMI game as a generalized Nash equilibrium problem. For continuous SPMI games, we further show that such a GNEP can be written as a linear complementarity problem (LCP) through the Karush-Kuhn-Tucker (KKT) optimality conditions. We then show that the resulting LCP has favorable properties, allowing the use of Lemke’s pivoting algorithm with guaranteed convergence to a solution (as opposed to a secondary ray).

We refer to the LCP approach as a centralized approach, in the sense that the game is purely viewed as a system of equilibrium conditions, and a general algorithm capable of solving the resulting system is applied. We also present decentralized algorithms based on best-response dynamics, which are applicable to both continuous and discrete SPMI games. While not necessarily more computationally efficient, decentralized algorithms indeed have several advantages over centralized algorithms. First, a centralized approach usually relies on the first-order optimality conditions of each agent’s optimization problem. Such conditions are not available in discrete games, where agents’ problems contain discrete variables. A decentralized approach can nevertheless be applied to discrete games, as each agent’s problem can be solved by an integer programming algorithm, without relying on explicit optimality conditions. Second, a decentralized algorithm may provide insight on how a particular equilibrium is achieved among agents’ strategic interactions. Such insight is particularly useful when multiple equilibria exist, as is the case for many GNEPs. It is known
(for example, [35]) that a game may possess unintuitive Nash equilibria that would never be a realistic outcome. Third, a decentralized algorithm can naturally lead to multithreaded implementations that can take advantage of a high performance computing environment. In addition, different threads in a multithreaded implementation may be able to find different equilibria of a game, making such an algorithm particularly suitable for computationally quantifying the average efficiency loss of noncooperative strategies.

In the following discussion, we first present the GNEP formulation of SPMI games under continuous interdiction. We then reformulate the GNEP as an LCP and analyze the properties of the LCP formulation. Finally we present the decentralized algorithms for SPMI games under both discrete and continuous interdiction formulated as GNEPs.

4.1 Dual GNEP formulation

In (9), each agent’s objective function \( \theta^f(x^f, x^{-f}) \) involves its adversary’s shortest path problem, which can be written as an integer program as follows:

\[
\theta^f(x^f, x^{-f}) = \left\{ \begin{array}{ll}
\min & \sum_{(u,v) \in A} z^f_{uv} d_{uv}(x^f, x^{-f}) \\
\text{s.t.} & \sum_{\{v \in V | (u,v) \in A\}} z^f_{uv} - \sum_{\{v \in V | (v,u) \in A\}} z^f_{vu} = \begin{cases} 
1 & \text{if } u = s^f \\
0 & \text{if } u \neq s^f, t^f \\
-1 & \text{if } u = t^f 
\end{cases} \\
z^f_{uv} \in \{0, 1\} \quad \forall (u,v) \in A
\end{array} \right., (11)
\]

where the binary variables \( z^f_{uv} \) represents whether an arc \((u,v) \in A\) is in the shortest \(s^f-t^f\) path. Although the inner minimization problem is an integer program with binary variables, the constraint matrix is totally unimodular (e.g., [43]), rendering the integer program equivalent to its linear programming relaxation. Therefore, once the interdictors’ variables \((x_1, \ldots, x_F)\) are fixed, we can use linear programming duality to transform the inner minimization problem to a maximization problem [27] and reformulate agent \(f\)’s optimization problem (1) as:

\[
\begin{align*}
\text{maximize} & \quad y^f_{t^f} - y^f_{s^f} \\
\text{subject to} & \quad y^f_v - y^f_u \leq d_{uv}(x^f, x^{-f}) \quad \forall (u,v) \in A, \\
& \quad \sum_{(u,v) \in A} c^f_{uv} x^f_{uv} \leq b^f, \\
& \quad x^f \in X^f, \\
& \quad y^f_v \geq 0 \quad \forall v \in V.
\end{align*}
\] (12)

It is also known (see, for example, [6, 30]) that at optimality, the term \(y^f_u - y^f_s\) is equal to the length of the shortest \(s^f-t^f\) path in the aftermath network. This is the reason why we are able to restrict the \(y^f\) variables to be non-negative. In addition, it also allows us to restrict the \(y^f\) variables to be integral if the underlying network data is integral, since at optimality all path lengths would also be integral. Moreover, as we show below, it also allows us to bound the \(y^f\) variables.

When interdiction is continuous, the largest possible length in the aftermath network for any arc is bounded by the largest interdiction possible on that arc. Keeping the budgetary constraints in mind, we observe that the maximum length of any arc \((u,v) \in A\) in the aftermath network is bounded by

\[
d^0_{uv} + F \cdot \max_{f \in F, (u,v) \in A} \left\{ \frac{b^f}{c^f_{uv}} \right\}.
\]
Therefore, the length of any path in the aftermath network is bounded above by

\[ \bar{Y} = \sum_{(u,v) \in A} d_{uv}^0 + |A| \max_{f \in \mathcal{F}, (u,v) \in A} \left\{ \frac{b_f}{c_{uv}^f} \right\} . \]

On the other hand, when interdiction is discrete, the length of any path in the aftermath network is bounded above by \( \bar{Y} = \sum_{(u,v) \in A} (d_{uv}^0 + e_{uv}) \).

Since only the differences \( y_v^f - y_u^f \) across arcs \( (u,v) \) are relevant to the formulation (12), we may always replace \( y_u^f \) by \( y_u^f - y_s^f \) for each \( u \in V \) to obtain a feasible solution with equal objective value. Therefore we can then add the constraints \( 0 \leq y_u^f \leq \bar{Y} \) for all \( u \in V \) to the problem (12) to obtain an equivalent formulation of a SPMI game, where each agent \( f \in \mathcal{F} \) solves the following problem:

\[
\begin{align*}
\text{maximize} \quad & y_{tf}^f - y_{sf}^f \\
\text{subject to} \quad & y_{v}^f - y_{u}^f \leq d_{uv}(x^f, x^{-f}) \quad \forall (u, v) \in A, \\
& \sum_{(u,v) \in A} c_{uv}^f x_{uv}^f \leq b_f^f, \\
& 0 \leq y_u^f \leq \bar{Y} \quad \forall u \in V, \\
& x^f \in X^f .
\end{align*}
\]

When analyzing the SPMI game from a centralized decision-making perspective, we assume that the global obstruction function is utilitarian, i.e., the sum of the shortest \( s^f-t^f \) path lengths over all the agents \( f \in \mathcal{F} \). We also assume that the resources are pooled among all the agents, resulting in a common budgetary constraint. Thus the centralized problem for SPMI games can be given as follows:

\[
\begin{align*}
\text{maximize} \quad & \sum_{f \in \mathcal{F}} (y_{tf}^f - y_{sf}^f) \\
\text{subject to} \quad & y_{v}^f - y_{u}^f \leq d_{uv}(x^f, x^{-f}) \quad \forall (u, v) \in A, f \in \mathcal{F}, \\
& \sum_{f \in \mathcal{F}} \sum_{(u,v) \in A} c_{uv}^f x_{uv}^f \leq \sum_{f \in \mathcal{F}} b_f^f, \\
& 0 \leq y_u^f \leq \bar{Y} \quad \forall u \in V, f \in \mathcal{F}, \\
& x^f \in X^f \quad \forall f \in \mathcal{F}.
\end{align*}
\]

Since \( y^f \) is bounded for all \( f \in \mathcal{F} \), the feasible set for (14) is compact. Thus a globally optimal solution exists regardless of whether \( x^f \) is continuous or discrete for all \( f \in \mathcal{F} \). In the continuous case, Weierstrass's extreme value theorem applies since all the functions are continuous and the \( x^f \) variables are bounded due to the non-negativity and budgetary constraints. In the discrete case, there are only a finite number of values that the \( x^f \) variables can take.

The formulation (13) gives us some insight into the structure of strategic interactions among agents in a SPMI game. Note that in formulation (13), the objective function for each agent \( f \in \mathcal{F} \) only depends on variables indexed by \( f \) (in particular, \( y_{sf}^f \) and \( y_{tf}^f \)). However, the constraint set for each agent \( f \) is parametrized by other agents’ variables \( x^{-f} \), which leads to a generalized Nash equilibrium problem.

As before, let \( \mathcal{F} = \{1, \ldots, F\} \) denote the set of agents. Let the scalar-valued function \( \theta^f(\chi^f, \chi^{-f}) \) be the utility function of agent \( f \in \mathcal{F} \), which is a function of all the agents’ actions \( (\chi^f, \chi^{-f}) \). The feasible action space of agent \( f \in \mathcal{F} \) is a set-valued mapping \( \Xi^f(\chi^{-f}) \) with dimension \( n_f \) (in a regular Nash equilibrium problem, each agent’s feasible action space is a fixed set). Let \( n := \sum_{f \in \mathcal{F}} n_f \). Then \( \Xi^f(\cdot) \) is
a mapping from \( \mathbb{R}^{(n-n_f)} \) to \( \mathbb{R}^{n_f} \). Parametrized by the other agents’ decisions \( \chi^{-f} \), each agent \( f \in F \) in a GNEP solves the following problem:

\[
\begin{align*}
\text{maximize} & \quad \theta^f(\chi^f, \chi^{-f}) \\
\text{subject to} & \quad \chi^f \in \Xi^f(\chi^{-f}).
\end{align*}
\]

It is straightforward to see how the SPMI game in \( (13) \) translates into a GNEP problem: for all \( f \in F \),

\[
\chi^f = (x^f, y^f),
\]

\[
\theta^f(\chi^f, \chi^{-f}) = y^f_t - y^f_s,
\]

\[
\Xi^f(\chi^{-f}) = \begin{cases} \\
\chi^f = (x^f, y^f) \\
y^f_u - y^f_u \leq d_{uv}(x^f, x^{-f}) \quad \forall (u, v) \in A, \\
\sum_{(u,v) \in A} c^f_{uv}x^f_{uv} \leq b^f, \\
0 \leq y^f_u \leq \bar{Y} \quad \forall u \in V, \\
x^f \in X^f
\end{cases}
\]

Note that \( \chi = (\chi^1, \ldots, \chi^F) \in \mathbb{R}^n \), where \( n = F \times (|V| + |A|) \).

To formally define a Nash equilibrium to a GNEP, we let \( \Omega(\chi) \) denote the Cartesian product of the feasible sets of each agent corresponding to decisions \( \chi = (\chi^1, \ldots, \chi^F) \); that is,

\[
\Omega(\chi) := \Xi^1(\chi^{-1}) \times \Xi^2(\chi^{-2}) \times \cdots \times \Xi^F(\chi^{-F}).
\]

For a simultaneous-move GNEP with each agent solving problem \( (15) \), a generalized Nash equilibrium is defined as follows:

**Definition 1.** A vector \( \chi_N = (\chi^1_N, \ldots, \chi^F_N) \in \Omega(\chi_N) \) is a pure-strategy generalized Nash equilibrium (PGNE) if for each agent \( f \in F \),

\[
\theta^f(\chi^f_N, \chi^{-f}_N) \geq \theta^f(\chi^f, \chi^{-f}_N), \quad \forall \chi^f \in \Xi^f(\chi^{-f}_N).
\]

Based on the above definitions and discussions, it is easy to see that if \((x, y)\) is an equilibrium to a SPMI game formulated as a GNEP using both primal and dual variables \( (16) \), then \( x \) must be an equilibrium to the SPMI game using only the primal variables \( (9) \). Such a relationship is formally stated below.

**Proposition 3.** Suppose that \( \chi = (x, y) \in \mathbb{R}^{F \times (|A|+|V|)} \) is a PGNE to the GNEP where each agent solves \( (16) \). Then \( x \) is a PNE to the SPMI game where each agent solves \( (9) \).

For the remainder of the paper, we will mainly use the GNEP formulation, as our focus shifts from establishing theoretical properties of SPMI equilibria to computing such an equilibrium. Under continuous interdiction, the GNEP formulation can be further reformulated as an LCP, as we show below. On the other hand, under discrete interdiction, we show that certain classes of SPMI games admit provably convergent decentralized algorithms. In this case, we sequentially solve agents’ problems \( (16) \) using an integer programming solver.

### 4.2 Linear Complementarity Formulation

Before presenting the LCP formulation for the SPMI game, we introduce some basic notation and definitions. Formally, given a vector \( q \in \mathbb{R}^d \) and a matrix \( M \in \mathbb{R}^{d \times d} \), a linear complementarity problem \( \text{LCP}(q, M) \) consists of finding a decision variable vector \( w \in \mathbb{R}^d \) such that

\[
w \geq 0,
\]

\[
Mw + q \geq 0, \quad w^T(Mw + q) = 0.
\]
\[ q + Mw \geq 0, \]  
\[ w^T(q + Mw) = 0. \]  
\[ (20) \]

The LCP \((q, M)\) is said to be feasible if there exists a \(w \in \mathbb{R}^d\) that satisfies \((19)\) and \((20)\). Any \(w\) satisfying \((21)\) is called complementary. If \(w\) is both feasible and complementary, it is called a solution of the LCP, and the set of such solutions is denoted by \(\text{SOL}(q, M)\). The LCP is said to be solvable if it has a solution. A thorough exposition of the theory and algorithms for LCPs can be found in [13].

Consider now the SPMI game with continuous interdiction, introduced in Section 2.2, where agent \(f\)'s optimization problem is given in \((12)\). When the interdiction decisions of the agents \(f', f \neq f\) are fixed, \((12)\) is a linear program (LP). In this case, the KKT conditions are both necessary and sufficient for a given feasible solution to be optimal.

In order to present the LCP reformulation in a more compact form, we introduce the following notation. Let \(|V| = n\) and \(|A| = m\). Denote by \(G\) the arc-node incidence matrix of the graph \(G\). Further let \(I\) denote an identity matrix, and \(0\) be a vector or a matrix of all zeros, of appropriate dimensions, respectively. The objective function coefficients for the LP \((12)\) are given by the vector \(\phi^f \in \mathbb{R}^{m+n}\), defined as follows:

\[ \phi^f = \begin{bmatrix} 0_m \\ \nu^f \end{bmatrix}, \quad \text{where} \quad \nu^f = \begin{cases} 1 & \text{if } u = s^f \\ 0 & \text{if } u \neq s^f, t^f \\ -1 & \text{if } u = t^f \end{cases}. \]

The right hand sides for the constraints given by the vector \(r^f(x-f) \in \mathbb{R}^{m+1}\):

\[ r^f(x-f) = \begin{bmatrix} -d^0 \\ -b^f \end{bmatrix} - \sum_{f' \in F, f' \neq f} \begin{bmatrix} I_m & 0_{m \times n} \\ 0_{m \times m} & I_n \end{bmatrix} \begin{bmatrix} x^{f'} \\ y^{f'} \end{bmatrix}. \]

The left hand sides for the constraints are given by the matrix \(A^f \in \mathbb{R}^{(m+1) \times (m+n)}:\n
\[ A^f = \begin{bmatrix} I_m & G \\ -c^f & 0_n \\ \end{bmatrix}. \]

Using this notation, the LP \((12)\) can be restated as follows:

\[
\begin{align*}
\text{minimize} & \quad \phi^f x^f \\
\text{subject to} & \quad A^f \begin{bmatrix} x^f \\ y^f \end{bmatrix} \geq r^f(x-f), \\
& \quad \begin{bmatrix} x^f \\ y^f \end{bmatrix} \geq 0.
\end{align*}
\]

Let the dual variables for the LP \((22)\) be \((\lambda^f, \beta^f, \upsilon^f)\), where \(\lambda^f\) are the multipliers for the arc potential constraints, \(\beta^f\) the multiplier for the budgetary constraint and \(\upsilon^f\) the multipliers for the non-negativity constraints. The KKT conditions for \((22)\) are given by the following system.

\[
\begin{align*}
\begin{bmatrix} x^f \\ y^f \end{bmatrix} & \leq A^f \begin{bmatrix} \lambda^f \\ \beta^f \end{bmatrix} \geq 0, \\
0 & \leq \begin{bmatrix} x^f \\ y^f \end{bmatrix} \perp \upsilon^f \geq 0, \\
\phi^f - A^f \begin{bmatrix} \lambda^f \\ \beta^f \end{bmatrix} - \upsilon^f & = 0.
\end{align*}
\]

\[ (23) \]
The KKT system (23) can be rewritten in the following form:

\[ \begin{align*}
    v_f &= \phi_f - A_f^T \begin{bmatrix} \lambda_f \\ \beta_f \end{bmatrix} \geq 0, \\
    x_f^0 &\geq 0, \\
    y_f^T v_f &= 0, \\
    t_f &= -r_f(x_f) + A_f \begin{bmatrix} x_f \\ y_f \end{bmatrix} \geq 0, \\
    \lambda_f^T \geq 0, \\
    t_f^T \begin{bmatrix} \lambda_f \\ \beta_f \end{bmatrix} &= 0.
\end{align*} \tag{24} \]

In this form, it is easy to recognize that for a fixed value of \( x_f \), the KKT system of all the other agents, which has the following specific form:

\[ q_f(x_f) = \begin{bmatrix} \phi_f \\ -r_f(x_f) \end{bmatrix} \quad \text{and} \quad M_f = \begin{bmatrix} 0_{(m+n) \times (m+n)} & -A_f^T \\ A_f & 0_{(m+1) \times (m+1)} \end{bmatrix}. \tag{25} \]

The decision variable vector for the LCP is the vector of combined decision variables \( w_f := (x_f, y_f, \lambda_f, \beta_f)^T \). Each agent’s KKT system (24) is parametrized by the collective decisions of other agents.

Now by stacking all agents’ KKT systems together, the resulting model is itself an LCP, which can be seen from the following algebraic manipulation. First, consider the following system obtained from (24) by expanding \( r_f(x_f) \):

\[ \begin{align*}
    v_f &= \phi_f - A_f^T \begin{bmatrix} \lambda_f \\ \beta_f \end{bmatrix} \geq 0, \\
    x_f^0 &\geq 0, \\
    y_f^T v_f &= 0, \\
    t_f &= \begin{bmatrix} d^0_f \\ b_f \end{bmatrix} + A_f \begin{bmatrix} x_f \\ y_f \end{bmatrix} + \sum_{f' \in F, f' \neq f} \begin{bmatrix} I_m \\ 0_{m \times n} \\ 0_{n \times m} \\ 0_{n \times 1} \end{bmatrix} \begin{bmatrix} x_{f'} \\ y_{f'} \end{bmatrix} \geq 0, \\
    \lambda_f^T \geq 0, \\
    t_f^T \begin{bmatrix} \lambda_f \\ \beta_f \end{bmatrix} &= 0. \tag{26} \\
\end{align*} \]

Now introduce a matrix \( \tilde{M}_f \) to represent the interactions between agent \( f \)’s decision variables \( (x_f, y_f) \) and the KKT system of all the other agents, which has the following specific form:

\[ \tilde{M}_f = \begin{bmatrix}
    0_{m \times m} & 0_{m \times n} & 0_{m \times n} & 0_{m \times 1} \\
    0_{n \times m} & 0_{n \times n} & 0_{n \times n} & 0_{n \times 1} \\
    I_m & 0_{m \times n} & 0_{n \times m} & 0_{n \times 1} \\
    0_{1 \times m} & 0_{1 \times n} & 0_{m \times m} & 0_{1 \times 1} \\
\end{bmatrix}. \tag{27} \]

Using this notation, the stacked KKT system (26) for agents \( f = 1, \ldots, F \) can be formulated as an LCP(\( q, M \)), with the vector \( q \) and matrix \( M \) given as follows:

\[ q = (q^1, q^2, \ldots, q^F)^T, \quad \text{where} \quad \tilde{q}_f = (\phi_f, d^0, b_f)^T, \tag{28} \]

and

\[ M = \begin{bmatrix}
    M^1 & \tilde{M}_2 & \tilde{M}_3 & \cdots & \tilde{M}_F \\
    M^1 & M^2 & M^3 & \cdots & M^F \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    M^1 & M^2 & \cdots & M^{F-1} & M^F \\
\end{bmatrix}. \tag{29} \]

Due to the equivalence between an agent \( f \)’s optimization problem (12) and its KKT system (24), the above LCP(\( q, M \)) is equivalent to the corresponding (continuous) SPMI game in the sense that a candidate point \( (\chi^1, \chi^2, \ldots, \chi^F) \), where \( \chi^f = (x_f, y_f) \), is an equilibrium to the SPMI game if and only if there exist associated Lagrangian multipliers such that they together solve the LCP(\( q, M \)).
Methods for solving LCPs fall broadly into two categories: (i) pivotal methods such as Lemke’s algorithm, and (ii) iterative methods such as splitting schemes and interior point methods. The former class of methods are finite when applicable, while the latter class converge to solutions in the limit. In general, the applicability of these algorithms depends on the structural properties of the matrix \( M \). In the following analysis, we show that LCP \((q, M)\) for the SPMI game, as defined in (28) and (29), possesses two properties that allow us to use Lemke’s pivotal algorithm: (i) the matrix \( M \) is a copositive matrix, and (ii) \( q \in (\text{SOL}(0, M))^* \).

We first show that \( M \) is copositive. Recall that a matrix \( M \in \mathbb{R}^{d \times d} \) is said to be copositive if \( x^T M x \geq 0 \) for all \( x \in \mathbb{R}^d_+ \).

**Lemma 1.** \( M \) defined as in (29) is copositive.

**Proof.** Let \( w \in \mathbb{R}^{2m+n+1}_+ \). Using the block structure of \( M \) given in (29), \( w^T M w \) can be decomposed as follows.

\[
w^T M w = \sum_{f=1}^{F} w^T M^f w^f + \sum_{f=1}^{F} \sum_{f' \neq f}^F w^T \bar{M}^f w^{f'}.
\]

(30)

We analyze the terms in the two summations separately. First consider \( w^T M^f w^f \) for any agent \( f \). Let the dual variables \((\lambda^f, \beta^f)\) be collectively denoted by \( \delta^f \). We have

\[
w^T M^f w^f = \begin{bmatrix} \chi^f T & \delta^f T \end{bmatrix} \begin{bmatrix} 0 & -A^f T \\ A^f & 0 \end{bmatrix} \begin{bmatrix} \chi^f \\ \delta^f \end{bmatrix} = -\chi^T A^f \delta^f + \delta^T A^f \chi^f = 0.
\]

(31)

Now consider any term of the form \( w^T \bar{M}^{f'} w^{f'} \):

\[
w^T \bar{M}^{f'} w^{f'} = \begin{bmatrix} x^{f'} T & y^{f'} T & \chi^{f'} T & \beta^{f'} T \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ L_m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x^{f'} \\ y^{f'} \\ \chi^{f'} \\ \beta^{f'} \end{bmatrix} = \chi^{f'} T x^{f'}.
\]

(32)

Combining (31) and (32) we obtain

\[
w^T M w = \sum_{f=1}^{F} \sum_{f' \neq f}^F \lambda^{f'} x^{f'}.
\]

(33)

Since \( \lambda^{f'} \)'s and \( x^{f'} \)'s are the elements of \( w \), \( w \geq 0 \) clearly implies that \( w^T M w \geq 0 \).

We now show that \( q \in (\text{SOL}(0, M))^* \).

**Lemma 2.** Let the vector \( q \) and the matrix \( M \) be as defined in (28) and (29) respectively. Then \( q \in (\text{SOL}(0, M))^* \).

---

1Given a set \( K \in \mathbb{R}^d \), the set \( K^* \) denotes the dual cone of \( K \); i.e. \( K^* = \{ y \in \mathbb{R}^d : y^T x \geq 0, \forall x \in K \} \).
First note that $\text{SOL}(0, M) \neq \emptyset$ for any $M$, since 0 is always a solution to LCP$(0, M)$. Now consider a $w \in \text{SOL}(0, M)$; i.e. $0 \leq w \perp 0 + Mw \geq 0$. We prove that $q^Tw \geq 0$. Observe that $q^Tw$ can be decomposed as follows:

$$q^Tw = \sum_{f=1}^{F} q^fT^Tw^f = \sum_{f=1}^{F} \left( \phi^fT^T \left[ x^f \right] + d^fT^T \lambda^f + b^f \beta^f \right)$$

$$= \sum_{f=1}^{F} \left[ (y^f_{s^f} - y^f_{t^f}) + d^fT^T \lambda^f + b^f \beta^f \right].$$

(34)

The last two terms in the last equality above, $d^fT^T \lambda^f$ and $b^f \beta^f$, are non-negative for $f = 1, \ldots, F$ because $w \in \text{SOL}(0, M)$ implies that $\lambda^f, \beta^f \geq 0$, and by assumption $d^f, b^f \geq 0$ for each $f = 1, \ldots, F$.

Now we focus on the first term in the last equality of (34): $\sum_{f=1}^{F} (y^f_{s^f} - y^f_{t^f})$. First since $Mw \geq 0$, $w^f$ must solve the system (26) for $f = 1, \ldots, F$, with $\phi^f$, $d^0$ and $b^f$ all set at 0 (as $\phi^f$, $d^0$ and $b^f$ are the components of the vector $q$ in the LCP$(q, M)$, as defined in (28); and in LCP$(0, M)$, $q = 0$). In this case, considering the primal feasibility of $w^f$, we obtain the following:

$$\sum_{a \in A} c^f_a x^f_a \leq 0$$

$$y^f_u - y^f_v + \sum_{f=1}^{F} x^f_{u,v} \geq 0 \quad \forall (u, v) \in A$$

(35)

for $f = 1, \ldots, F$.

Recall that $c^f_a \geq 0$ for all $a \in A$ and $f = 1, \ldots, F$ by assumption. Therefore, (35) implies that $x^f = 0$ for any agent $f$. It is easy to see that in this case, we must have

$$y^f_u - y^f_v \geq 0 \quad \forall (u, v) \in A, \text{ for } f = 1, \ldots, F.$$  

(36)

Now consider any $s^f$-$t^f$ path $p$. By assumption, there must be at least one such path for each agent $f$. By summing up the inequalities (36) over the arcs in the path $p$, we obtain the desired result. In other words,

$$\sum_{(u, v) \in p} y^f_u - y^f_v = y^f_{s^f} - y^f_{t^f} \geq 0.$$  

(37)

Summing up over the agents, we thus have shown that $q^Tw \geq 0$ for any $w \in \text{SOL}(0, M)$. \hfill $\square$

With Lemma 1 and 2 we can apply the following result from Cottle et al. [13].

**Theorem 1** (13, Theorem 4.4.13). If $M$ is copositive and $q \in (\text{SOL}(0, M))^*$, then Lemke’s method will always compute a solution, if the problem is nondegenerate.

As discussed earlier, the LCP approach is not applicable for discrete SPMI games due to the lack of necessary and sufficient optimality conditions. In the following we develop a decentralized approach that works for both discrete and continuous SPMI games.
4.3 Gauss-Seidel Algorithm (Algorithm 1)

We first present the basic form of a best response based algorithm. The idea is simple: starting with a particular feasible decision vector \( \chi_0 = (\chi_1^0, \chi_2^0, \ldots, \chi_F^0) \in \Omega(\chi_0) \), solve the optimization problem of a particular agent, say, agent 1, with all of the other agents’ actions fixed. Assume an optimal solution exists to this optimization problem, and denote it as \( \chi_1^* \). The next agent, say, agent 2, solves its own optimization problem, with the other agents’ actions fixed as well, but with \( \chi_0^1 \) replaced by \( \chi_1^* \). Such an approach is often referred to as a diagonalization scheme or a Gauss-Seidel iteration, and for the remainder of this paper we use the latter name to refer to this simple best-response approach.

Consider applying the Gauss-Seidel iteration to a GNEP, with each agent solving the optimization problem (15), denoted as \( P(\chi^- f) \). The Gauss-Seidel iterative procedure is presented in Algorithm 1 below. Note

Algorithm 1 Gauss-Seidel Algorithm for a GNEP

Initialize. Choose \( \chi_0 = (\chi_1^0, \ldots, \chi_F^0) \) with \( \chi_f^0 \in \Xi_f(\chi^- f) \) \( \forall f \in F \). Set \( k \leftarrow 0 \).

Step 1:

for \( f = 1, 2, \ldots, F \) do

set \( \chi_{k,f} \leftarrow (\chi_1^{k+1}, \ldots, \chi_{f-1}^{k+1}, \chi_{f}^k, \ldots, \chi_F^k) \);

solve \( P(\chi^- f) \) to obtain an optimal solution \( \chi_{k,f}^* \); if \( \theta_f(\chi_{k,f}^*, \chi^- f) > \theta_f(\chi_{k}^f, \chi^- f) \) then set \( \chi_{k+1}^f \leftarrow \chi_{k,f}^* \); else set \( \chi_{k+1}^f \leftarrow \chi_{k}^f \);

end if

end for

set \( \chi_{k+1} \leftarrow (\chi_1^{k+1}, \ldots, \chi_F^{k+1}) \).

set \( k \leftarrow k + 1 \).

if \( \chi_k \) satisfies termination criteria, then STOP; else GOTO Step 1.

that updates in agent \( f \)'s decisions occur at iteration \( k \) only if there is a strict increase in the agent’s payoff at the iteration. The algorithm can be directly applied to compute an equilibrium of a SPMI game with discrete interdiction. For finite termination, we fix a tolerance parameter \( \epsilon \) and use the following stopping criterion:

\[ \| \chi_k - \chi_{k-1} \| \leq \epsilon. \]  

(38)

**Proposition 4.** Suppose that the Gauss-Seidel algorithm (Algorithm 2) is applied to the SPMI game with discrete interdiction, and the termination criterion \( (38) \) is used with \( \epsilon < 1 \). If the algorithm terminates at \( \chi_k \), then \( \chi_k \) is an equilibrium to this problem.

**Proof.** Since the variables \( \chi_k \) are integral for discrete interdiction problems, choosing \( \epsilon < 1 \) for the termination criterion will ensure that the algorithm terminates only when successive outer iterates are equal. Consequently, by the assumption, \( \chi_{k-1} = \chi_k \) at termination. This also implies that \( \chi_{k-1,f}^- = \chi_k^- f \) for \( f = 1, \ldots, F \). By construction of \( \chi_k \), we must then have

\[ \chi_k^f = \arg \min_{\chi^f \in \Xi_f(\chi_k^- f)} \theta_f(\chi^f, \chi_k^- f). \]

Clearly, \( \chi_k \) must then be an equilibrium. \( \square \)

Even though there is no guarantee that the algorithm will in fact converge, we note that in the discrete case, it is possible to detect when the algorithm fails to converge. Recall that \( \Xi_f(\chi^- f) \subseteq K_f \) for each agent.
\( f \in F \), where \( K^f \) is defined below.

\[
K^f = \left\{ (x^f, y^f) \mid \sum_{(u,v) \in V} e^f_{uv} x^f_{uv} \leq b^f, \quad 0 \leq y^f_u \leq \bar{Y} \quad \forall u \in V \right\}.
\] (39)

Clearly, the set \( \prod_{f=1}^F K^f \) is finite. Any intermediate point \( \chi_k \) generated by Algorithm 2 must certainly satisfy the budgetary constraints on \( x^f_k \) and the bound constraints on \( y^f_k \) for each agent \( f \). Therefore \( \chi_k \in \prod_{f=1}^F K^f \). In other words, the set of possible points \( \chi_k \) generated by Algorithm 2 lies in a finite set. This means that if the algorithm fails to converge, it must generate a sequence that contains at least one cycle. The existence of such cycles in non-convergent iterate paths can then be used to detect situations in which the algorithm fails to converge.

Proposition 4 is likely the best one can do for general SPMI games under discrete interdiction. However, for the subclass of such games with common source-target pairs, we can in fact prove that the best response dynamics always terminates in a NE in a finite number of steps.

**Proposition 5.** Consider a SPMI game with discrete interdiction with common source-target pairs, and assume that the initial arc lengths \( d \) and arc extensions \( e \) are integral. Suppose that Algorithm 2 is applied to such a problem, and the termination criteria (38) is used with \( \epsilon < 1 \). Then the algorithm will terminate finitely at an equilibrium.

**Proof.** Denote the common source node as \( s \), and the common target node as \( t \). The set of joint feasible strategies in \( x \) under the given assumptions is a finite set. Moreover, all the agents attempt to minimize the common objective, namely the \( s-t \) path length. Note that at any iteration \( k \) at which an update occurs for any agent’s decision, there must then be a strict increase in the \( s-t \) path length. Thus there cannot exist cycles in the sequence \( \{\chi_k\} \). Furthermore, since the set of joint feasible strategies is finite, the sequence must terminate at some point \( \chi^* \). It is easy to show that \( \chi^* \) must be an equilibrium (cf. Proposition 4). \( \square \)

### 4.4 Regularized Gauss-Seidel Algorithm (Algorithm 2)

One disadvantage of the “naïve” Gauss-Seidel algorithm described above is that for continuous GNEPs, it can fail to converge to an equilibrium. However, Facchinei et al. [21] showed that under certain assumptions, we can overcome this issue by adding a regularization term to the individual agent’s problem solved in a Gauss-Seidel iteration.

The regularized version of the optimization problem for agent \( f \in F \) is

\[
\begin{align*}
\text{maximize} & \quad \theta^f(\chi^f, \chi^{-f}) - \tau \left\| \chi^f - \bar{\chi}^f \right\|^2 \\
\text{subject to} & \quad \chi^f \in \Xi^f(\chi^{-f}),
\end{align*}
\] (40)

where \( \tau \) is a positive constant. Here the regularization term is evaluated in relation to a candidate point \( \bar{\chi}^f \). Note that the point \( \bar{\chi}^f \) and the other agents’ decision variables \( \chi^{-f} \) are fixed when the problem (40) is solved. We refer to problem (40) as \( R(\chi^{-f}, \bar{\chi}^f) \). The regularized Gauss-Seidel procedure, herein referred to as Algorithm 2, is simply Algorithm 1, except that \( R(\chi_k^{-f}, \chi_k^f) \) is solved in each iteration \( k \) instead of \( P(\chi_k^{-f}, \chi_k^f) \).

This version of the algorithm, along with its convergence proof, was originally presented in [21] to solve GNEPs with shared constraints. The difficulty here that prevents us from showing convergence lies in the fact that we are dealing with GNEPs with non-shared constraints. As a result, any intermediate points resulting from an agent’s best responses need not to be feasible in the other agents’ problems. Consequently,
Algorithm 2: Gauss-Seidel Algorithm for a GNEP

Initialize. Choose \( \chi_0 = (\chi_1^0, \ldots, \chi_F^0) \) with \( \chi_f^0 \in \Xi^f(\chi_0^0) \) \( \forall f \in F \). Set \( k \leftarrow 0 \).

Step 1: 
for \( f = 1, 2, \ldots, F \) do 
    Set \( \chi_{k+f} \leftarrow (\chi_1^{k+1}, \ldots, \chi_{k}^{f-1}, \chi_1^f, \ldots, \chi_F^k) \);
    Solve \( R(\chi_{k+f}^f, \chi_f^f) \) to obtain an optimal solution \( \chi_{k}^* \);
    Set \( \chi_{k+f}^f \leftarrow \chi_{k}^* \);
end for
Set \( \chi_{k+1} \leftarrow (\chi_1^{k+1}, \ldots, \chi_F^{k+1}) \).
Set \( k \leftarrow k + 1 \).
if \( \chi_k \) satisfies termination criteria, then STOP; else GOTO Step 1.

we use Algorithm 2 only as a heuristic algorithm to solve SPMI games under continuous interdiction. Nevertheless, we can show that if Algorithm 2 converges, then the resulting point is an equilibrium to the SPMI game.

Proposition 6. Let \( \{\chi_k\} \) be the sequence generated by applying Algorithm 2 to the SPMI problem under continuous interdiction, wherein each agent solves the regularized version of (13). Suppose \( \{\chi_k\} \) converges to \( \bar{\chi} \). Then \( \bar{\chi} \) is an equilibrium to the SPMI problem.

The proof of this proposition is almost identical to that of Theorem 4.3 in Facchinei et al. [21]. However, we do want to point out one key difference in the proof. In Proposition 6, we need to assume that the entire sequence \( \{\chi_k\} \) converges to \( \bar{\chi} \). This is a strong assumption in the sense that it also requires that all the intermediate points \( \chi_{k+f} \) in Algorithm 2 to converge to \( \bar{\chi} \), a fact key to proving that \( \bar{\chi} \) is indeed an equilibrium. In contrast, for GNEPs with shared constraints, this assumption may be weakened because the intermediate points \( \chi_{k+f} \) and therefore the cluster points of the sequence generated by the algorithm are guaranteed to be feasible. The complete proof is presented in Appendix A.

Similar to discrete SPMI games, the convergence of Algorithm 2 is guaranteed for continuous-interdiction SPMI games with common source-target pairs. The key fact that allows us to prove this stronger result is that by dropping the dependence of the variables \( y \) on the agents \( f \in F \), any unilateral deviation in the shared variables \( y \) results in a solution that remains feasible in the other agents’ optimization problems. The convergence result is formally stated below.

Proposition 7. Consider applying Algorithm 2 to the SPMI problem under continuous interdiction with common source-target pairs, where each agent solves the regularized version of (13). Let \( \{\chi_k\} \) be the sequence generated by the algorithm. If \( \bar{\chi} \) is a cluster point of this sequence, then it also solves the SPMI problem.

5 Numerical Results

We use the algorithms presented in the previous section to study several instances of SPMI games. The decentralized algorithms were implemented in MATLAB R2010a with CPLEX v12.2 as the optimization solver. The LCP formulation for the SPMI game with continuous interdiction was solved using the MATLAB interface for the complementarity solver PATH [22]. Computational experiments were carried out on a desktop workstation with a quad-core Intel Core i7 processor and 16 GHz of memory running Windows 7.

In the implementation of the decentralized algorithm, for SPMI games with discrete interdiction, we used Algorithm 2 with \( \tau = 0 \). For SPMI games with continuous interdiction, we followed a strategy of trying the “naïve” Gauss-Seidel algorithm – i.e. Algorithm 2 with \( \tau = 0 \) – first. If this version failed to
converge in 1000 outer iterations, we then set \( \tau \) to a strictly positive value and used the more expensive regularized Gauss-Seidel algorithm.

**Computing Equilibria**

First, we applied the algorithm to Example 2 in Section 3.2 which is a SPMI game with continuous interdiction. In particular, the network is given in Figure 2 and there are 2 agents: agent 1 has an adversary with source node 1 and target node 5, and agent 2 has an adversary with source node 1 and target node 6. Both agents have an interdiction budget of 1. The initial arc lengths are 0, and the interdiction costs are equal for both agents and are given as the arc labels in Figure 2 with \( \epsilon = 2 \). We set the regularization parameter \( \tau = 0.01 \). We were able to obtain a solution within an accuracy of \( 10^{-6} \) in 3 outer iterations.

Furthermore, we obtained multiple Nash equilibria by varying the starting point of the algorithm. All the equilibria obtained resulted in the same shortest path lengths for each agent. Some of the equilibria obtained are given in Table 3. The column \( x_0 \) represents the starting interdiction vector for each agent, the columns \( x_N^1 \) and \( x_N^2 \) give the equilibrium interdiction vectors for agents 1 and 2, respectively. The seven components in the vectors of \( x_0, x_N^1 \) and \( x_N^2 \) represent the interdiction actions at each of the seven arcs in Figure 2 with the arcs being ordered as follows: first, the top horizontal arcs (1, 2) and (2, 3), then the vertical arcs (1, 4), (2, 5) and (3, 6), and finally the bottom horizontal arcs (4, 5) and (5, 6). The remaining two columns in Table 3, \( p_1 \) and \( p_2 \), give the shortest path lengths for agents 1 and 2 respectively, at the equilibrium \( \chi_N \).

<table>
<thead>
<tr>
<th>( x_0 )</th>
<th>( x_N^1 )</th>
<th>( x_N^2 )</th>
<th>( p_1 )</th>
<th>( p_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0, 0, 0, 0, 0, 0)</td>
<td>(0, 0, 0, 0, 0, 0, 0)</td>
<td>(0, 0, 0, 0, 0, 0, 0)</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
<tr>
<td>(0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2)</td>
<td>(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)</td>
<td>(0, 0, 0.0697, 0.0697, 0.6667, 0.6667)</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
<tr>
<td>(0.0, 0.0, 0.0, 0.3, 0.3, 0.3, 0.3)</td>
<td>(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)</td>
<td>(0, 0, 0.3167, 0.3167, 0.6667, 0.6667)</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
<tr>
<td>(0.3, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)</td>
<td>(0.0, 0.6667, 0.6667, 0.6667, 0.6667)</td>
<td>(0, 0, 0.0167, 0.0167, 0.6667, 0.6667)</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
<tr>
<td>(0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25)</td>
<td>(0.0, 0.625, 0.625, 0.625, 0.625)</td>
<td>(0, 0, 0.0417, 0.0417, 0.6667, 0.6667)</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
<tr>
<td>(0.0, 0.15, 0.15, 0.15, 0.15, 0.15, 0.15)</td>
<td>(0.0, 0.425, 0.425, 0.425, 0.425)</td>
<td>(0, 0, 0.2417, 0.2417, 0.6667, 0.6667)</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
<tr>
<td>(0.15, 0.15, 0.15, 0.15, 0.15, 0.15, 0.15)</td>
<td>(0.0, 0.575, 0.575, 0.575, 0.575)</td>
<td>(0, 0, 0.0917, 0.0917, 0.6667, 0.6667)</td>
<td>0.6667</td>
<td>0.6667</td>
</tr>
</tbody>
</table>

![Figure 3: Network structure for SPMI Example 4](image)

**Example 4.** To test the algorithm on larger-scale problems, we expanded the instance in Example 2 to larger network sizes and numbers of agents. For \( F \) agents, the graph contains \( 2(F + 1) \) vertices with the edges as shown in Figure 3. The source vertex for all agents is \( a_1 \). The target vertex for a given agent \( f \) is \( b_{f+1} \). The initial arc lengths are all assumed to be zero. The interdiction costs are the same for all the agents and are...
given as the arc labels in Figure 3. All the agents have an interdiction budget of 1. The cost parameter $\epsilon$ is chosen as 2. For discrete interdiction on these graphs, the arc extensions are assumed to be length 1.

The running time and iterations required to compute equilibria for these instances are summarized in Table 4. The first four columns in the table give the number of outer iterations and runtime for Algorithm 2 over these instances with continuous interdiction. The results indicate that the running time for the centralized Lemke’s method increases monotonically with the problem size. However, the running time for the decentralized method depends not just on the problem size but also on the number of outer iterations. In general, there is no correlation between these two parameters. Indeed the algorithm is observed to converge in relatively few iterations even for some large problem instances. This is in stark contrast to the rapid increase in running time observed for the LCP approach as problem size increases.

It must be noted that the order in which the individual agent problems are solved in the Gauss-Seidel algorithm plays an important role. Indeed, the algorithm failed to converge for certain orderings of the agents, but succeeded in finding equilibria quickly for the same instance with other orderings. For instance, for a network of size 25, solving the agent problems in their natural order $\{1, 2, \ldots, 25\}$ resulted in the failure of the “naïve” version of the algorithm to converge even after 1000 outer iterations. However, with a randomized agent order, the algorithm converged in as few as 13 iterations. It is encouraging to note that for the same agent order that resulted in the failure of the naïve version, the regularized method converged to a GNE within 394 outer-iterations with a runtime of 28 wall-clock seconds.

### Table 4: Number of iterations and running times for SPMI Example 4.

<table>
<thead>
<tr>
<th># Agents</th>
<th>Continuous Interdiction</th>
<th>Discrete Interdiction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Decentralized LCP</td>
<td>Decentralized</td>
</tr>
<tr>
<td></td>
<td># Iters</td>
<td>Runtime (s)</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0.0205</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.0290</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>0.1103</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>0.0723</td>
</tr>
<tr>
<td>25</td>
<td>13</td>
<td>0.2609</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0.4070</td>
</tr>
<tr>
<td>35</td>
<td>10</td>
<td>0.3605</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>1.7485</td>
</tr>
<tr>
<td>45</td>
<td>12</td>
<td>0.6601</td>
</tr>
<tr>
<td>50</td>
<td>12</td>
<td>0.7981</td>
</tr>
</tbody>
</table>

**Computation of Efficiency Losses**

Using the decentralized algorithm and its potential to find multiple equilibria by starting at different points, we numerically study the efficiency loss of decentralized interdiction strategies in SPMI games. We focus first on Example 4 with the underlying network represented in Figure 3. We first show that in general, the worst-case price of anarchy cannot be bounded from above. We do so by demonstrating that given any candidate upper bound on the worst-case price of anarchy, we can construct an instance that invalidates the bound.

Consider the specific instance of the problem as depicted in Figure 3. Recall that there are $F$ agents and the source-target pair for agent $f$ is $(a_1, b_{f+1})$. Note that all paths for all agents contain either the arc $(a_1, a_2)$ or the arc $(a_1, b_1)$. Then one feasible solution to the centralized problem is for each agent to interdict both these arcs by $1/(2 + \epsilon)$ for a total cost of 1. In this case, the length of both arcs become $F/(2 + \epsilon)$, giving a shortest path length of $F/(2 + \epsilon)$ for each agent. Note that this is not an equilibrium.
solution as agent 1 can deviate unilaterally to interdict arcs \((a_1, b_1)\) and \((a_2, b_2)\) by \(1/2\) to obtain a shortest path length of \((F + \epsilon/2)/(2 + \epsilon)\).

A Nash equilibrium to this instance is given by the following solution. Agent \(f\) interdicts the vertical arcs \((a_1, b_1), \ldots, (a_f, b_f)\) by \(1/(f(f + 1))\) and the arc \((a_{f+1}, b_{f+1})\) by \(f/(f + 1)\). Each agent then has a shortest path length of \(F/(F + 1)\). Note that all the \(s^f-t^f\) paths are of equal length for every agent. Therefore diverting any of the budget to any vertical arcs will result in unequal path lengths and a shorter shortest path for any agent. Obviously, diverting the budget to interdict any of the horizontal arcs is cost inefficient because of their higher interdiction cost at \(1 + \epsilon\). Thus no agent has an incentive to deviate from this solution.

We now have a feasible solution to the centralized problem that has an objective value of \(F/(2 + \epsilon)\) for each agent, and a Nash equilibrium that has an objective value of \(F/(F + 1)\) for each agent. Therefore, by its definition in \((4)\), the worst-case price of anarchy for the SPMI game depicted in Figure \(3\) must be at least \((F + 1)/(2 + \epsilon)\).

The observation above implies that given any fixed candidate upper bound on the worst-case price of anarchy for the general class of SPMI games, under continuous interdiction, we can easily compute a tuple \((F, \epsilon)\), which gives us an instance of the problem that breaks the bound.

Using the regularized Gauss-Seidel algorithm we also compute lower bounds on the worst-case price of anarchy and average efficiency losses for the same network topology with varying number of agents. The instances we consider are obtained by varying \(\epsilon\) uniformly in the range of \((1.5, 10)\). For the purpose of comparison, the numerical results are plotted in Figure \(4\) below. Note that the average-case efficiency loss is much lower than the worst-case price of anarchy. For the particular graph structure under consideration, we observe that the average efficiency loss grows at a much lower rate than the worst-case efficiency loss. However this observation cannot be generalized to other graph structures and such patterns may only be discernible by applying a decentralized computational framework as the one we presented.

![Figure 4: Efficiency loss with respect to the number of agents.](image)

**Example 5.** We further tested the decentralized algorithms for continuous interdiction on random graphs to study average efficiency losses of equilibria of SPMI games on networks with different topologies. For the random graphs we generated, the input parameters include the number of vertices and the density of
a graph, which is the number of arcs divided by the maximum possible number of arcs. The number of agents was chosen randomly from the interval \((0, |V| / 2)\), and one such number is chosen per vertex set size. Source-target pairs were chosen at random for each interdictor. Fixing the vertex set, we populated the arc set by successively generating source-target paths for the agents until the desired density was reached. We thus ensured connectivity between the source-target pairs for each agent. Costs, initial arc lengths and interdiction budgets were chosen from continuous uniform distributions. Arc interdiction costs were assigned uniformly in the range \([1, 5]\). The budget for each agent \(f\) was chosen uniformly from the interval \([b_f / 10, b_f / 2]\), where \(b_f = \sum_{a \in A} c^{f}_a\). The initial length of each arc was chosen uniformly from \([1, 5]\).

For each combination of vertex set size, the number of agents, and graph density, we generated 25 random instances by drawing values from the uniform distributions described above for the various network parameters. For each instance, we used 10 different random permutations of the agents to run the decentralized algorithms in an attempt to compute multiple equilibria. The lower bound on the price of anarchy for the game was computed as the worst case efficiency loss over these 25 instances. The average efficiency loss over these instances was also computed. The results are summarized in Table 5. Our experiments indicate that the average efficiency loss and the worst-case price of anarchy tend to grow as the number of vertices and number of agents increases; on the other hand, these measures of efficiency loss sometimes do not appear to be monotonically increasing or decreasing with respect to the density of the underlying network.

<table>
<thead>
<tr>
<th># Vertices</th>
<th># Agents</th>
<th>Density</th>
<th>Avg. Run Time (s)</th>
<th># Avg Iters.</th>
<th>a.e.l</th>
<th>p.o.a</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>0.25</td>
<td>0.0037</td>
<td>3</td>
<td>1.3133</td>
<td>1.5561</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0.5</td>
<td>0.0038</td>
<td>3</td>
<td>1.3265</td>
<td>1.9529</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0.75</td>
<td>0.0040</td>
<td>3</td>
<td>1.5099</td>
<td>2.3829</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>0.25</td>
<td>0.0065</td>
<td>4</td>
<td>1.5366</td>
<td>2.2078</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>0.5</td>
<td>0.0176</td>
<td>11</td>
<td>1.4538</td>
<td>2.3114</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>0.75</td>
<td>0.0132</td>
<td>8</td>
<td>1.4273</td>
<td>2.1342</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>0.25</td>
<td>0.0263</td>
<td>11</td>
<td>1.7091</td>
<td>2.9246</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>0.5</td>
<td>0.0939</td>
<td>33</td>
<td>1.7524</td>
<td>2.7904</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>0.75</td>
<td>0.1267</td>
<td>42</td>
<td>1.5695</td>
<td>2.1425</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>0.25</td>
<td>0.1269</td>
<td>34</td>
<td>2.1907</td>
<td>3.2885</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>0.5</td>
<td>0.2087</td>
<td>43</td>
<td>1.8523</td>
<td>2.7906</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>0.75</td>
<td>0.5416</td>
<td>100</td>
<td>1.7967</td>
<td>2.3782</td>
</tr>
<tr>
<td>25</td>
<td>7</td>
<td>0.25</td>
<td>0.7167</td>
<td>105</td>
<td>2.5631</td>
<td>4.8788</td>
</tr>
<tr>
<td>25</td>
<td>7</td>
<td>0.5</td>
<td>1.9564</td>
<td>207</td>
<td>2.3022</td>
<td>5.5794</td>
</tr>
<tr>
<td>25</td>
<td>7</td>
<td>0.75</td>
<td>1.8476</td>
<td>158</td>
<td>1.9884</td>
<td>2.4423</td>
</tr>
</tbody>
</table>

### 6 Conclusions and Future Work

In this work, we introduced decentralized multi-interdictor games and gave formulations for one such class of games – shortest path multi-interdictor games. We analyzed the theoretical properties of SPMI games: in particular, we gave conditions for the existence of equilibria and examples where multiple equilibria exist. Specifically, we proved the existence of equilibria for general SPMI games under continuous interdiction. On the other hand, for the discrete counterpart, we provide an example where a pure-strategy equilibrium does not exist. However, for the subclass of problems with common source-target pairs, we are able to provide an existence guarantee.

We also showed that the SPMI game under continuous interdiction is equivalent to a linear complementarity problem, which can be solved by Lemke’s algorithm. This constitutes a convergent centralized method to solve such problems. We also presented decentralized heuristic algorithms to solve SPMI games under both continuous and discrete interdiction. Finally, we used these algorithms to numerically evaluate the worst case and average efficiency loss of SPMI games.
There are other classes of network interdiction games that can be studied using the same framework we have developed, where the agents’ obstruction functions are related to the maximum flow or minimum cost flow in the network. Establishing theoretical results and studying the applicability of the decentralized algorithms to other classes of decentralized network interdiction games are natural and interesting extensions of this work.

In our study of SPMI games, we also made the assumption that the games have complete information structure; that is, the normal form of the game – the set of agents, agents’ feasible action spaces, and their objective functions – is assumed to be common knowledge to all agents. In addition, we made the implicit assumption that all input data are deterministic. However, data uncertainty and lack of observability of other agents’ preferences or actions are prevalent in real-world situations. For such settings, we need to extend our work to accommodate games with exogenous uncertainties and incomplete information.

One might also be interested in designing interventions to reduce the loss of efficiency resulting from decentralized control. This leads to the topic of mechanism design. Such a line of work also defines a very important and interesting future research direction.

Acknowledgement

This work was partially supported by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-12-1-0275.

A Proof of Proposition 6

We prove the proposition in two steps. We first show that $\bar{\chi}$ is feasible to each player’s problem. Since by assumption $\chi_k \to \bar{\chi}$, we must have $\chi_{k}^f \to \bar{\chi}^f$ and

$$\lim_{k \to \infty} \|\chi_{k+1}^f - \chi_k^f\| = 0, \quad \forall f \in \mathcal{F}. \quad (41)$$

By construction of $\chi_{k,f}$, (41) implies that

$$\lim_{k \to \infty} \chi_{k,f} = \bar{\chi}, \quad \forall f \in \mathcal{F}. \quad (42)$$

Consider $\chi_{k,f+1} = (\chi_{k+1}^1, \ldots, \chi_{k+1}^f, \chi_k^{f+1}, \chi_k^F)$. By Step 1 of Algorithm 2, we must have

$$\chi_{k+1}^f \in \Xi_f(\chi_{k,f+1}). \quad (43)$$

Note that by (41) and (42), $\chi_{k+1}^f \to \bar{\chi}^f$ and $\chi_k^{f+1} \to \bar{\chi}^{f+1}$. The set $\Xi_f(\chi_{k,f})$ is defined by linear inequalities parametrized by $\chi_{k,f}$. Thus we may utilize continuity properties of this set valued mapping, and take limits on (43) to obtain

$$\bar{\chi}^f \in \Xi_f(\bar{\chi}^{-f}). \quad (44)$$

In other words, $\bar{\chi}$ is feasible for every agent’s optimization problem (15).

We complete the proof by showing that for each agent $f \in \mathcal{F}$

$$\theta^f(\bar{\chi}^f, \bar{\chi}^{-f}) \geq \theta^f(\chi^f, \bar{\chi}^{-f}), \quad \forall \chi^f \in \Xi_f(\bar{\chi}^{-f}).$$

For the purpose of establishing a contradiction, suppose that there is an agent $\bar{f}$ and a vector $\bar{\xi}^\bar{f} \in \Xi_f(\bar{\chi}^{-f})$ such that

$$\theta^f(\bar{\chi}^\bar{f}, \bar{\chi}^{-\bar{f}}) < \theta^f(\bar{\xi}^\bar{f}, \bar{\chi}^{-\bar{f}}).$$
Let $d^f = (\xi^f - \bar{\chi}^f)$. Then by the subdifferentiability inequality for concave functions we must have

$$\theta^f(\bar{\chi}^f, \chi^f; d^f) > 0.$$  \hfill (45)

Our proof relies on constructing a contradiction to (45). To do so, we first construct a sequence $\xi^f_k$ that is feasible to agent $\hat{f}$’s problem at the $k$-th iteration, such that $\xi^f_k \to \xi^f$.

Using the linearity of the functions that define the set valued mapping $\Xi(f)$ we can show its inner semicontinuity relative to its domain (cf. \cite{chapter} Chapter 5). Because $\bar{\chi}^{-f} \in \text{dom}(\Xi(f))$, we then have

$$\liminf_{\xi^{-f} \to \bar{\chi}^{-f}} \Xi(\xi^{-f}) \supseteq \Xi(\bar{\chi}^{-f}),$$  \hfill (46)

where the limit in (46) is given by the following:

$$\liminf_{\xi^{-f} \to \bar{\chi}^{-f}} \Xi(\xi^{-f}) = \left\{ u^f \mid \forall \xi_k^{-f} \to \bar{\chi}^{-f}, \exists u_k^f \to u^f \text{ with } u_k^f \in \Xi(\xi_k^{-f}) \right\}. \hfill (47)$$

By assumption, $\xi^f \in \Xi(f(\bar{\chi}^{-f}))$. By (42) we also have $\chi_k^{-f} \to \bar{\chi}^{-f}$. Equation (47) then allows us to construct a sequence $\xi_k^f \in \Xi(\chi_k^{-f})$ such that $\xi_k^f \to \xi^f$ as $k \to \infty$.

Denote by $\Phi^f$ the regularized objective function for agent $f$’s subproblem. In other words,

$$\Phi^f(\chi^f, \chi^{-f}, z) = \theta^f(\chi^f, \chi^{-f}) - \tau \left\| \chi^f - z \right\|^2.$$

We then have

$$\Phi^f(\chi^f, \chi^{-f}, z; d^f) = \theta^f(\chi^f, \chi^{-f}; d^f) - 2\tau(\chi^f - z)^T d^f.$$

Note that $\chi_{k+1}^f$ is obtained by solving the problem $\mathcal{R}(\chi_k^{-f}, \chi_k^f)$. In other words, $\chi_{k+1}^f$ maximizes $\Phi^f(\cdot, \chi_k^{-f}, \chi_k^f)$ over the set $\Xi(\chi_k^{-f})$. Applying first order optimality conditions, setting $z = \chi^f_k$ and $d^f = \xi_k^f - \chi_{k+1}^f$, we obtain the following.

$$\Phi^f(\chi_{k+1}^f, \chi_k^{-f}, \chi_k^f; (\xi_k^f - \chi_{k+1}^f)) = \theta^f(\chi_{k+1}^f, \chi_k^{-f}; (\xi_k^f - \chi_{k+1}^f))$$

$$+ 2\tau(\chi_{k+1}^f - \chi_k^f)(\xi_k^f - \chi_{k+1}^f) \leq 0.$$  \hfill (48)

Passing to the limit $k \to \infty$, $k \in K$ and using (42) we obtain

$$0 \geq \theta^f(\bar{\chi}^f, \chi^{-f}; (\bar{\xi}^-f - \bar{\chi}^f)),$$

which contradicts (45).

\[\square\]

References


