Abstract

In this work, we introduce decentralized network interdiction games, which model the interactions between multiple interdictors with differing objectives operating on a common network. These games can be seen as Generalized Nash Equilibrium Problems (GNEPs) with non-shared constraints. We focus on decentralized shortest path network interdiction games, and establish the existence of equilibria for these games under both discrete and continuous interdiction strategies. While computing an equilibrium of a GNEP is challenging in general, we show that under continuous interdiction actions the game can be reformulated 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 actions. Finally, we establish theoretical bounds on the worst-case efficiency loss of equilibria in these games, and use our decentralized algorithms to empirically 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 [53], preventing nuclear smuggling [39] and planning tactical air strikes [29]. Interdiction models have also found applications in other areas such as controlling the spread of pandemics [4] and defending attacks on computer communication networks [50].

Traditionally, interdiction problems have been analyzed from a centralized perspective. In other words, 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. For instance, a supervising body, in control of multiple agents in a common system, may assign each agent to an adversary of interest. Each agent is then responsible for computing and implementing its own interdiction strategy against the designated adversary. Other situations may involve multiple independent agents, such as security agencies of different countries, trying to achieve a common goal on a shared network. Without any coordination between the agents, one might expect that a decentralized interdiction
strategy may be inefficient compared to one determined by a central decision maker. This paper is focused on modeling and analyzing such settings and the inefficiencies that may arise.

In this paper, we introduce \textit{decentralized network interdiction (DNI) 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 \textit{decentralized shortest path interdiction (DSPI) games}. We investigate various properties of equilibria in DSPI games, including their existence and uniqueness, and propose algorithms to compute equilibria of these games. Using these algorithms, we also conduct empirical studies on the efficiency loss of equilibria in the DSPI game in comparison 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. [10] and Smith and Lim [50].

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 [7, 43, 57], in which selfish agents route traffic through a congested network, and congestion games [46], a generalization of routing games. Some examples include [5, 8, 11, 13, 27, 47, 48, 51]. 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 [1–3, 15, 20]. The inefficiency of equilibria has been studied in other games as well, such as facility location games [54], scheduling games [36], and resource allocation games [33, 34]. 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 [12, 52], research in this direction has not received much attention.

One potential reason for the lack of attention paid to decentralized network interdiction games may be that such games belong to the class of generalized Nash equilibrium problems (GNEPs), 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 [15] and Rosen [45], rigorous theoretical and algorithmic treatments of GNEPs only began in recent years [22]. Several techniques have been proposed to solve GNEPs, including penalty-based approaches [23, 28, 42], variational-inequality-based approaches [41], Newton’s method [19], projection-like methods [59], and relaxation approaches [37, 53]. Most of the work on GNEPs has focused on games with shared constraints, mainly due to their tractability [21, 30]. 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.

One of the major contributions of this work is that we show the GNEPs derived from DSPI games to have special mathematical structure. First, we show that a DSPI game (either under discrete or continuous interdiction) always admits a potential function (whose exact definition will be provided in Section 3.2). With this result, we are able to prove the existence of equilibria for DSPI games. Second, for continuous interdiction DSPI games, 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. In addition, we present decentralized algorithms for finding equilibria of DSPI games. These algorithms are based on the well-known best-response dynamics (or Gauss-Seidel iterative) approach. While
such an approach is only a heuristic method (as convergence to a Nash equilibrium cannot be shown, even with the potential function property), we obtain encouraging empirical results for the performance of the method on several classes of network structures. We also use these algorithms on some instances of DSPI games to empirically quantify the average-case efficiency loss when interdicting agents behave selfishly. Such 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 DNI games and DSPI games. In Section 3, we present the main theoretical results of the paper, including an analysis of the existence and uniqueness of equilibria in DSPI games, as well as a proof that these games admit potential functions. In Section 4 we investigate algorithms for solving DSPI games. We describe a centralized algorithm based on a linear complementarity formulation, as well as decentralized algorithms for computing equilibria of DSPI games. We also give results of our computational experiments with these algorithms on instances of DSPI games. Finally, in Section 5, we provide some concluding remarks.

2 Decentralized Network Interdiction Games

2.1 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 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; in other words, $x^{-f} = (x^1, \ldots, x^{f-1}, x^{f+1}, \ldots, x^F)$. 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 $f \in \mathcal{F}$ faces a total interdiction budget of $b^f > 0$. 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 $x_{uv}^f$ units is $c_{uv}^f x_{uv}^f$. We assume that $c_{uv}^f > 0$ for all $(u, v) \in A$ and $f \in \mathcal{F}$.
The optimization problem for each agent $f \in \mathcal{F}$ is:

$$\begin{align*}
\text{maximize} & \quad \theta_f^I(x_f^I, x_{-f}^I) \\
\text{subject to} & \quad \sum_{(u,v) \in \mathcal{A}} c_{uv}^f x_{uv}^f \leq b_f^I, \\
& \quad x_f^I \in X_f^I,
\end{align*}$$

(1)

where the objective function $\theta_f^I$ 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 network interdiction (DNI) game. The obstruction function $\theta_f^I$ can capture various types of interdiction problems. Typically $\theta_f^I$ is the (implicit) optimal value function of the adversary’s network optimization problem parametrized by the agents’ decisions, which usually minimizes flow cost or path length subject to flow conservation, arc capacity and side constraints.

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 $\theta_c^e(x^1, \ldots, x^F)$ represent the global obstruction function for a given interdiction strategy $(x^1, \ldots, x^F)$. The central planner’s problem corresponding to the DNI game (1) is then as follows:

$$\begin{align*}
\text{maximize} & \quad \theta_c^e(x^1, \ldots, x^F) \\
\text{subject to} & \quad \sum_{f \in \mathcal{F}} \sum_{(u,v) \in \mathcal{A}} c_{uv}^f x_{uv}^f \leq \sum_{f \in \mathcal{F}} b_f^I, \\
& \quad x_f^I \in X_f^I \quad \forall f \in \mathcal{F}.
\end{align*}$$

(2)

We refer to (2) as the centralized problem, and focus primarily on when the global obstruction function is utilitarian; that is,

$$\theta_c^e(x^1, \ldots, x^F) := \sum_{f \in \mathcal{F}} \theta_f^I(x_f^I, x_{-f}^I).$$

As mentioned earlier, one of the goals of this work is to quantify the inefficiency of an equilibrium of a DNI 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 price of anarchy.

Formally speaking, let $\mathcal{N}_I$ be the set of all equilibria corresponding to a specific instance $I$. (In the context of DNI 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 global optimal solution to the centralized problem (2). Then the price of anarchy of the instance $I$ is defined as

$$p(I) := \max_{(x_1^*, \ldots, x_F^*) \in \mathcal{N}_I} \frac{\theta_c^e(x_1^*, \ldots, x_F^*)}{\theta_c^e(x_1^I, \ldots, x_F^I)}.$$  

(3)

Let $\mathcal{I}$ be the set of all instances of a game. We assume implicitly that for all $I \in \mathcal{I}$, the set $\mathcal{N}_I$ is nonempty and a global optimal solution to the centralized problem exists. By convention, $p$ is set to 1 if the worst equilibrium as well as the global optimal solution to the centralized problem both have zero objective value. If the worst equilibrium has a zero objective value while the global optimum is nonzero, $p$ is set to 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:

$$w.p.o.a := \sup_{I \in \mathcal{I}} p(I).$$  

(4)
Since we wish to study properties of a class of games such as DNI games, rather than a particular instance of a game, we are more interested in the worst-case price of anarchy. However, there are two major difficulties associated with such an efficiency measure. First, it is well-known that the worst-case price of anarchy may 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. Indeed most of the related research has focused on identifying classes of games where such bounds may be derived. In this work, we show how our proposed decentralized algorithms can be used to empirically study the average-case efficiency loss (denoted by $a.e.l$).

Let $I'$ denote a finite set such that $I' \subset I$, and let $|I'|$ denote the cardinality of the set $I'$. Then

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

(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, such as maximum flow interdiction. To start with models that are both theoretically and computationally tractable, we focus on decentralized shortest-path interdiction games, which we describe in detail next.

### 2.1.1 Decentralized Shortest Path Interdiction Games

As the name suggests, decentralized shortest path interdiction (DSPI) games involve players 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 for the general decentralized network interdiction game described in Section 2.1. Each agent $f \in F$ has a target node $t_f \in V$ which 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 arc lengths on the network: the decision variable $x_{uv}^f$ represents the contribution of agent $f \in F$ towards lengthening arc $(u, v) \in A$. The arc length $d_{uv}(x^1, \ldots, x^F)$ 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 continuous: in particular,

$$X^f := \{ x^f \in \mathbb{R}^{|A|} : x_{uv}^f \geq 0 \ \forall (u, v) \in A \}$$

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

$$d_{uv}(x^1, \ldots, x^F) = d_{uv}^0 + \sum_{f \in F} x_{uv}^f \ \forall (u, v) \in A,$$

(6)

where $d_{uv}^0 > 0$ is the initial length of arc $(u, v)$. We make the positivity assumption on initial arc lengths without loss of generality, since we may always pad the arc lengths without changing the shortest paths. In this scenario, $x_{uv}^f$ captures how much agent $f$ extends the length of arc $(u, v)$.

The second type of interdiction is discrete: in this case,

$$X^f := \{ x^f \in \mathbb{R}^{|A|} : x_{uv}^f \in \{0, 1\} \ \forall (u, v) \in A \}$$

and the arc lengths in the aftermath network are

$$d_{uv}(x^1, \ldots, x^F) = d_{uv}^0 + e_{uv} \max_{f \in F} x_{uv}^f \ \forall (u, v) \in A,$$

(7)
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.

The optimization problem solved by each agent $f \in \mathcal{F}$ in a DSPI game is given by (1), where

$$
\theta^f(x^f, x^{-f}) := \begin{pmatrix}
\min_{z^f} & \sum_{(u,v) \in A} z^f_{uv} d_{uv}(x^f, x^{-f}) \\
\text{s.t.} & \sum_{v \in V} z^f_{uv} - \sum_{v \in V} z^f_{vu} = \begin{cases} 
1 & \text{if } u = s^f \\
0 & \text{if } u \neq s^f, t^f \\
-1 & \text{if } u = t^f 
\end{cases} \\
z^f_{uv} \in \{0, 1\} & \forall (u,v) \in A
\end{pmatrix}
$$

where binary variable $z^f_{uv}$ in (8) represents whether an arc $(u, v) \in A$ is in the shortest $s^f$-$t^f$ path. In other words, agent $f$’s optimization problem is a bilevel optimization problem, where the inner minimization problem (8) is its adversary’s shortest path problem. Although the inner minimization problem is an integer program with binary variables, it is well known that the constraint matrix is totally unimodular (e.g. [49]), rendering the integer program equivalent to its linear programming relaxation. Therefore, once the interdictors’ variables $(x^f_1, \ldots, x^f_F)$ are fixed, we can use linear programming duality to transform the inner minimization problem to a maximization problem [32] and reformulate agent $f$’s optimization problem (1) as:

$$
\max_{x^f, y^f} \quad y^f_{s^f} - y^f_{u^f} \\
\text{subject to} \quad y^f_u - y^f_s \leq d_{uv}(x^f, x^{-f}) & \forall (u, v) \in A,
$$

$$
\sum_{(u, v) \in A} c^f_{uv} x^f_{uv} \leq b^f, \\
x^f \in X^f.
$$

It is well known (see, for example, [6,35]) that at optimality, the term $y^f_u - y^f_{s^f}$ is equal to the length of the shortest $s^f$-$u$ path in the aftermath network. This fact has several important implications for problem (9). For instance, it 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, the maximum interdiction possible on any arc is bounded by

$$
F \cdot \max_{f \in \mathcal{F}, (u, v) \in A} \left\{ \frac{b^f}{c_{uv}} \right\}.
$$

As a result, the maximum length of any arc $(u, v) \in A$ in the aftermath network is bounded by

$$
d^0_{uv} + F \cdot \max_{f \in \mathcal{F}, (u, v) \in A} \left\{ \frac{b^f}{c_{uv}} \right\}.
$$

Therefore, the lengths of every path in the aftermath network are bounded above by

$$
M = \sum_{(u, v) \in A} d^0_{uv} + |A| F \cdot \max_{f \in \mathcal{F}, a \in A} \left\{ \frac{b^f}{c_{a}} \right\}.
$$
On the other hand, when interdiction is discrete, the length of any path in the aftermath network is bounded above by

$$M = \sum_{(u,v) \in A} (d_{uv}^0 + e_{uv}).$$

Since only the differences $y_{uv}^f - y_{uv}^{f'}$ across arcs $(u, v)$ are relevant to the formulation (9), we may always replace $y_{uv}^f$ by $y_{uv}^f - y_{uv}^{f'}$ for each $u \in V$ to obtain a feasible solution with equal objective value. Therefore we can then add the constraints $-M \leq y_{uv}^f \leq M$ for all $u \in V$ to the problem (9) to obtain an equivalent formulation of a DSPI game, where each agent $f \in F$ solves the following problem.

maximize $y_{vf}^f - y_{sf}^f$
subject to $y_{uv}^f - y_{uv}^{f'} \leq d_{uv}(x^f, x^{-f}) \quad \forall (u, v) \in A$,
$$\sum_{(u,v) \in A} c_{uv}x_{uv}^f \leq b_f,$$
$$-M \leq y_{uv}^f \leq M \quad \forall u \in V,$$ 
$$x^f \in X^f.$$

(10)

When analyzing the DSPI 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 $f \in F$. We also assume that the resources are pooled among all the agents, resulting in a common budgetary constraint. Thus the centralized problem for DSPI games can be given as follows:

maximize $x, y$
subject to $y_{uv}^f - y_{uv}^{f'} \leq d_{uv}(x^f, x^{-f}) \quad \forall (u, v) \in A, f \in F$,
$$\sum_{f \in F} \sum_{(u,v) \in A} c_{uv}x_{uv}^f \leq \sum_{f \in F} b_f,$$
$$-M \leq y_{uv}^f \leq M \quad \forall u \in V, f \in F,$$
$$x^f \in X^f \quad \forall f \in F.$$

(11)

Since $y^f$ is bounded for all $f \in F$, a globally optimal solution of (11) exists regardless of whether $x^f$ is continuous or discrete for all $f \in F$. 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.

3 Game Structure and Analysis

3.1 Generalized Nash Equilibrium Problems

The formulation (10) gives us some insight into the structure of strategic interactions among agents in a DSPI game. Note that in formulation (10), the objective function for each agent $f \in F$ 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 $x^{-f}$. This is an instance of a generalized Nash equilibrium problem (GNEP).
Formally speaking, consider the following simultaneous-move game with complete information\(^1\). 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 payoff or objective function of agent \(f \in \mathcal{F}\), which is a function of the actions or decisions \((\chi^f, \chi^{-f})\) of all the agents. The feasible action space of agent \(f \in \mathcal{F}\) is a set-valued mapping \(\Xi^f(\chi^{-f})\) with dimension \(n_f\). This dependence of an agent’s feasible action space on the actions of other agents is the key feature of a GNEP; in a regular Nash equilibrium problem, each agent’s feasible action space is a fixed set. Let \(\mathcal{F} = \sum_{f \in \mathcal{F}} n_f\). 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 \(f \in \mathcal{F}\) in a GNEP solves the following problem:

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

(12)

It is straightforward to see how the DSPI game in (10) translates into a GNEP problem: for all \(f \in \mathcal{F}\),

\[
\chi^f = (x^f, y^f), \\
\theta^f(\chi^f, \chi^{-f}) = (y^f_s - y^f_f), \\
\Xi^f(\chi^{-f}) = \left\{ \chi^f = (x^f, y^f) \mid \begin{array}{c} y^f_v - y^f_u \leq d_{uv}(x^f, x^{-f}) \quad \forall (u, v) \in A, \\
\sum_{(u, v) \in A} c^f_{uv} x^f_{uv} \leq b^f, \\
-M \leq y^f_f \leq M \quad \forall u \in V, \\
x^f_f \in X^f \end{array} \right\}.
\]

(13)

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

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,

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

(14)

For a simultaneous-move GNEP with each agent solving problem (12), 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 (GNE) if for each agent \(f \in \mathcal{F}\),

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

(15)

GNEPs in general are more challenging than regular Nash equilibrium problems, both theoretically and computationally. While results on the existence of (pure-strategy) generalized Nash equilibria have been established, few results exist on uniqueness of such equilibria, largely due to the fact that few GNEPs have a unique equilibrium [22]. We will rely on both existing analytic results and the special structure of DSPI games to determine the existence of equilibria and their uniqueness in these games in later subsections. Before that, however, we first show an important property of the DSPI games; namely, DSPI games as formulated in (10) admit potential functions. This property is extremely helpful in establishing the existence of equilibria.

\(^1\)A game is said to be simultaneous-move if the agents must make their decisions without being aware of the other agents’ decisions. A game has complete information if the number of players, their payoffs and their feasible action spaces are common knowledge to all the players.
3.2 Potential Functions and Potential Games

Potential games were first proposed by Rosenthal [46] in the context of congestion games. Roughly speaking, in a potential game, an agent’s decision to unilaterally improve its utility can be translated to an ascent direction of a potential function, and any global maximum of the potential function is equivalent to a Nash equilibrium of the original game. As a result, decentralized algorithms in the form of best- or better-response dynamics for potential games can be shown to converge to a Nash equilibrium under relatively general conditions.

Monderer and Shapley [38] formalized the concept of potential games and introduced various subclasses of such games. Perhaps the simplest subclass consists of games possessing exact potential functions, where any unilateral change in an agent’s objective function corresponds to a similar change of value in the potential function. Formally, exact potential functions for GNEPs are defined as follows.

Definition 2 ([38]). A continuous function $P : \mathbb{R}^n \to \mathbb{R}$ is said to be an exact potential function to a GNEP if for each $\chi \in \Omega(\chi)$ we have

$$\theta^f(\chi^f, \chi^{-f}) - \theta^f(\xi^f, \chi^{-f}) = P(\chi^f, \chi^{-f}) - P(\xi^f, \chi^{-f}) \quad \forall \xi^f \in \Xi^f(\chi^{-f})$$

for each agent $f \in \mathcal{F}$.

We will now show that DSPI games do in fact admit exact potential functions. Since we only deal with exact potential functions, we will refer to them henceforth simply as potential functions.

Proposition 3. The DSPI game in (10) admits a potential function.

Proof. Consider the DSPI game in (10) formulated as a GNEP as in (13), and function $P : \mathbb{R}^n \to \mathbb{R}$ where

$$P(\chi) := \sum_{f \in \mathcal{F}} (y_{t_f}^f - y_{s_f}^f).$$

(16)

This is simply the sum of the individual agents’ objective functions. It is easy to see that for each $f \in \mathcal{F}$, when $\chi^{-f}$ is fixed, $P$ and $\theta^f$ differ only by a constant. Therefore we have

$$\theta^f(\chi^f, \chi^{-f}) - \theta^f(\xi^f, \chi^{-f}) = P(\chi^f, \chi^{-f}) - P(\xi^f, \chi^{-f}) \quad \forall \xi^f \in \Xi^f(\chi^{-f})$$

for all $f \in \mathcal{F}$. Thus $P$ is a potential function for the DSPI game. $\square$

3.3 Existence of Equilibria

We first consider the existence of equilibria in the DSPI game when interdiction decisions are continuous. In this case each interdictor’s optimization problem (10), with the arc length function $d_{uv}(\cdot)$ defined as in (6), is a linear program. The following result regarding general convex GNEPs can therefore be applied here.

Theorem 4 (Ichiishi [31]). Given a simultaneous-move GNEP with each agent $f \in \mathcal{F}$ solving (12), assume that the following conditions hold:

(i) There exist nonempty, convex and compact sets $K^f \subseteq \mathbb{R}^{n_f}$ for each agent $f \in \mathcal{F}$ such that for every $\chi = (\chi^1, \chi^2, \ldots, \chi^F) \in \prod_{f=1}^F K^f$, $\Xi^f(\chi^{-f})$ is nonempty, closed and convex. In addition, $\Xi^f(\chi^{-f}) \subseteq K^f$ and $\Xi^f(\cdot)$ is both upper and lower semicontinuous as a point-to-set map.

In best-response dynamics, agents take turns solving their own optimization problem, while keeping other agents’ decisions fixed. In better-response dynamics, an agent may only find a decision that improves its objective function, not necessarily one that optimizes it.
(ii) For every agent \( f \in \mathcal{F} \), the function \( \theta^f(\cdot, \chi^{-f}) \) is quasi-concave on \( \Xi_f(\chi^{-f}) \).

Then a (pure-strategy) generalized Nash equilibrium exists.

To apply Theorem 4, the following result will be useful.

**Theorem 5 (Rockafellar and Wets [44]).** For every \( f \in \mathcal{F} \), suppose

\[
\Xi^f(\chi^{-f}) = \{ x^f \mid g_i^f(\chi^f, \chi^{-f}) \leq 0 \text{ for } i = 1, \ldots, m^f \}
\]

where \( g_1^f, \ldots, g_{m^f}^f \) are finite, continuous functions and \( g_i^f(\chi^f, \chi^{-f}), \ldots, g_{m^f}^f(\chi^f, \chi^{-f}) \) are convex in \( \chi^f \) for each \( \chi^{-f} \). If for \( \tilde{\chi}^{-f} \) there is a point \( \tilde{\chi}^f \) such that \( g_i^f(\tilde{\chi}^f, \tilde{\chi}^{-f}) < 0 \) for \( i = 1, \ldots, m^f \), then \( \Xi^f \) is continuous not only at \( \tilde{\chi}^{-f} \) but at every \( \chi^{-f} \) in some neighborhood of \( \tilde{\chi}^{-f} \).

We now use these results to show the existence of equilibria for the DSPI game.

**Proposition 6.** Under continuous interdiction, a generalized Nash equilibrium exists for the DSPI game (10).

**Proof.** Recall the representation of a DSPI game as a GNEP in (13). We show that conditions (i) and (ii) in Theorem 4 hold. First, condition (ii) in Theorem 4 is immediately apparent from (10) and (13) since the objective functions are linear.

To show condition (i) holds, we define the set \( K^f \) for each \( f \in \mathcal{F} \) as follows:

\[
K^f = \left\{ (x^f, y^f) \mid \sum_{(u,v) \in V} c_{uv}^f x_{uv}^f \leq b^f, \quad -M \leq y_u^f \leq M \quad \forall u \in V \right\}.
\]  

(17)

Clearly \( K^f \) is nonempty for all \( f \in \mathcal{F} \) as it contains the zero vector. It is also easy to see that \( \Xi^f(x^{-f}, y^{-f}) \subseteq K^f \) and is a polyhedron, and therefore closed and convex for any given \( (x^{-f}, y^{-f}) \). The set \( \Xi^f(x^{-f}, y^{-f}) \) is also nonempty, since we can construct a feasible solution by setting \( x_{uv}^f = 0 \) for all \( (u,v) \in A \) and \( y_u^f \) to be the length of the shortest path from \( s^f \) to \( u \) for all \( u \in V \). Now all that remains to be shown for condition (i) in Theorem 4 to hold are the continuity properties of the point-to-set mapping \( \Xi^f(\cdot) \).

To do so we use Theorem 5. We assume without loss of generality that the arc lengths \( d_{uv}^0 \), are positive, since we may always add a fixed constant to all of the arc lengths to achieve this without changing the shortest paths. Under these assumptions it is easily seen that setting all the \( x_{uv}^f \) and \( y_u^f \) variables to 0 will result in the required strict interior point. Note that the point \( (x^f, y^f) = 0 \) is a strict interior point to the set irrespective of the value of \( (x^{-f}, y^{-f}) \). Combined with the linearity of the constraints, this then allows us to directly apply Theorem 5 to claim the continuity of the point-to-set mapping \( \Xi^f(\cdot) \) in some neighborhood of \( (x^{-f}, y^{-f}) \), which certainly implies its continuity at \( (x^{-f}, y^{-f}) \). Since this property holds at any \( (x^{-f}, y^{-f}) \), the mapping \( \Xi^f(\cdot) \) is both lower and upper semi-continuous on its domain.

For discrete interdiction, the main difficulty is that the formulation (10) has both discrete and continuous variables. As a result it has a discontinuous objective function with an uncountable action space, which makes analysis difficult. However it is possible to work around this issue by noting that if the input data (in particular, the initial arc lengths \( d_{uv}^0 \) and the arc extension lengths \( e_{uv} \) for all \( (u,v) \in A \) are integral (or rational), then we may assume that the variables \( y^f \) are integral (rational) at equilibrium without loss of generality. This then allows us to use the potential function (16) to show existence of a generalized Nash equilibrium.

**Proposition 7.** If the initial arc lengths and the arc extension lengths for the DSPI game (10) are integral, then there exists a generalized Nash equilibrium under discrete interdiction.
Proof. We claim that solving (10) under discrete interdiction is equivalent to solving the following problem:

\[
\begin{align*}
\text{maximize} & \quad y^f - y^f_a \\
\text{subject to} & \quad y^f_v - y^f_u \leq d^0_{uv}(x^f, x^f -) \quad \forall (u, v) \in A, \\
& -M \leq y^f_u \leq M \quad \forall u \in V, \\
& \sum_{a \in A} c^f_a x^f_a \leq b^f, \\
& y^f_u \in \mathbb{Z} \quad \forall u \in V, \\
& x^f_a \in \{0, 1\}^{|A|} \quad \forall a \in A.
\end{align*}
\]

(18)

In problem (10), once the \(x^f\) variables are fixed, the \(y^f_u\) variables represent the shortest \(s-u\) path lengths on the aftermath network. Therefore if the initial arc lengths \(d^0_{uv}\) and the arc extension lengths \(e_{uv}\) are integral for all \((u, v) \in A\), then there exists an optimal solution to (10) under discrete interdiction in which \(y^f_u\) variables must also be integral. Thus problem (10) under discrete interdiction and problem (18) are equivalent in this sense.

A game with each agent solving problem (18) is a GNEP with a finite action space since both decision variables \(x^f\) and \(y^f\) are now integral and the feasible set is bounded; in other words, \(\Xi^f(x^f, y^f)\) is a finite set. We refer to this game as the discrete DSPI game. It is easy to see that the function \(P(\cdot)\) defined in (16) is also a potential function for the discrete DSPI game.

Denote by \(\bar{X}\) the set of all combinations of feasible decisions among the agents in \(F\) in the discrete DSPI game; in other words,

\[\bar{X} := \{\chi = (\chi^1, \ldots, \chi^F) | \chi \in \Omega(\chi)\}.\]

(19)

Note that in the discrete DSPI game, \(\bar{X}\) is also a finite set. Therefore the function \(P(\chi)\) achieves a maximum over the set \(\bar{X}\). Suppose the maximum is achieved at \(\hat{\chi}\). We claim that \(\hat{\chi}\) is in fact an equilibrium to the discrete DSPI game. Indeed for each agent \(f \in F\) we have that

\[\Theta^f(\hat{x}^f, \hat{x}^f -) - \Theta^f(\chi_f^f, \hat{x}^f -) = P(\hat{x}^f, \hat{x}^f -) - P(\chi_f^f, \hat{x}^f -) \geq 0 \quad \forall \chi_f^f \in \Xi^f(\hat{x}^f -).\]

The equality follows since \(P(\cdot)\) is an exact potential function and the inequality follows since \(\hat{x}\) maximizes \(P\) over \(\bar{X}\). Therefore, \(\hat{\chi}\) must be a GNE for the discrete DSPI game, and hence a GNE for the DSPI game under discrete interdiction.

Note that although any solution that maximizes the potential function over the set \(\bar{X}\) is an equilibrium to the original DSPI game, the reverse is not true; that is, an equilibrium to the DSPI game is not necessarily a maximizer of the potential function.

3.3.1 Uniqueness of equilibria

Establishing conditions under which a DSPI game has a unique equilibrium is quite difficult. However, it is easy to show that there exist simple instances of DSPI games for which multiple equilibria exist. We give several such examples below.

Example 1. Consider the following instance, based on the network in Figure 1. 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 1. Both agents have a budget of 1.
Consider the case when $\epsilon = 2$. One generalized Nash equilibrium occurs when agent 1 interdicts the arcs $(1, 4)$ and $(2, 5)$ by $1/2$ each, and agent 2 interdicts arcs $(1, 4)$ and $(2, 5)$ by $1/6$, and arc $(3, 6)$ by $2/3$. In this case both agents end up with a shortest path length of $2/3$. It is easy to see that any unilateral deviation will result in a smaller shortest path length for the deviating agent. In fact, 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 will be a generalized Nash equilibrium, and there is a continuum of such decision variable combinations.

**Example 2.** A variant of this instance under discrete interdiction exhibits some interesting properties. Consider the same instance, except under discrete interdiction, with all of the arc extension lengths are equal to 1, and the budget for each agent equal to $1 + \epsilon$ where $\epsilon$ is an integer that is at least 1. One possible equilibrium is for agent 1 to interdict the arc $(1, 4)$ and for agent 2 to interdict the arc $(1, 2)$. If $\epsilon = 1$, then this will result in a shortest path length of 1 for each agent. In addition, this solution corresponds to the solution that maximizes the potential function $P$ in (16) used in the proof of Proposition 7. Note that agent 1 does not use its entire budget. A similar equilibrium occurs when agent 1 interdicts arcs $(1, 4)$ and $(2, 5)$ and agent 2 interdicts arc $(3, 6)$. In this case agent 2 ends up with unused budget.

**Example 3.** A more interesting situation occurs when $\epsilon = 0$ and the budget is 1. In this case, the 2 agents can interdict at most 1 arc each. Agent 1 interdicting arc $(1, 4)$ and agent 2 interdicting arc $(1, 2)$ results in an equilibrium in which both agents have shortest path lengths of 1. The potential function value for this equilibrium is 2.

However, there exist other equilibria in which one agent is worse off than the other. For instance, suppose agent 1 interdicts the arc $(5, 6)$ and agent 2 interdicts $(3, 6)$: agent 1’s shortest path length is 0, and agent 2’s shortest path length is 1. Although agent 1’s shortest path length is now 0, it has no incentive to deviate since there is no possible unilateral deviation that would allow it to increase its shortest path length. A similar situation occurs when agent 1 interdicts $(2, 5)$ and agent 2 interdicts $(4, 5)$. Note that the potential function $P$ in (16) has a value of 1 for the latter two equilibria, which is strictly less than the value of 2 for the equilibrium described in the previous paragraph. It is also interesting to note that in this game, zero interdiction by both agents is also a generalized Nash equilibrium with a potential function value of 0.

### 4 Computing a Nash Equilibrium

In this section we focus on algorithms to compute equilibria of DSPI games. As discussed above, a DSPI game is a special case of a generalized Nash equilibrium problem. Computational methods to find an equilib-
rium for GNEPs include reformulations as quasi-variational inequalities [42], optimization reformulations using the Nikaido-Isoda function [18, 55, 56], direct methods using KKT systems [17] and penalty methods [23, 24], among others. We refer to the above methods as centralized algorithms, as they all attempt to find an equilibrium by tackling the game as a whole: for instance, by solving an equivalent variational inequality or complementarity problem. Such methods are usually computationally intensive.

Motivated by the observation that DSPI games admit potential functions, we also present decentralized algorithms based on best-response dynamics. Such decentralized algorithms have several advantages over centralized algorithms. First, the computational burden at each iteration is much smaller than with centralized algorithms, since only a single agent’s optimization problem is solved with others agents’ decisions fixed. Second, a decentralized algorithm may provide insight into how an 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 well-known (for example, [40]) that a game may possess unintuitive Nash equilibria that would never realistically be the outcome of the game. A centralized algorithm would not be able to distinguish between a meaningful and a meaningless equilibrium, and may end up computing such unintuitive equilibria. A decentralized algorithm, on the other hand, depicts how an equilibrium is achieved from a particular starting point through iterative interactions among agents, should the algorithm converge. Third, decentralized algorithms 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 decentralized strategies. Nevertheless, despite these favorable properties, best-response based algorithms suffer from a major drawback: it is difficult to theoretically prove these algorithms converge to equilibria for general classes of GNEPs.

In the following discussion, we propose solving the continuous DSPI game using a linear complementarity problem (LCP) reformulation. The reformulation is constructed using the Karush-Kuhn-Tucker (KKT) optimality conditions for each agent’s optimization problem. We show that the resulting LCP has favorable properties, allowing the use of Lemke’s pivoting algorithm.

### 4.1 Linear Complementarity Formulation

Before presenting the LCP formulation for the DSPI 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 LCP($q, M$) consists of finding a decision variable vector $w \in \mathbb{R}^d$ such that

\begin{align}
    w & \geq 0, \quad (20) \\
    q + Mw & \geq 0, \quad (21) \\
    w^T(q + Mw) & = 0. \quad (22)
\end{align}

The LCP($q, M$) is said to be feasible if there exists a $w \in \mathbb{R}^d$ that satisfies (20) and (21). Any $w$ satisfying (22) is called complementary. If $w$ is both feasible and complementary, it is called a solution of the LCP. In this case, we say $w \in \text{SOL}(q, M)$ to denote that $w$ is in the solution set for the LCP. The LCP is said to be solvable if it has a solution. A thorough exposition of the theory underlying LCPs and various algorithmic techniques to solve such problems can be found in [14].

Consider now the DSPI game with continuous interdiction, introduced in Section 2.1.1. We restate the
formulation (9) for the optimization problem of agent \( f \in F \) as follows:

\[
\begin{align*}
\text{minimize} & \quad y_f^f - y_{tf}^f \\
\text{subject to} & \quad y_u^f - y_v^f + x_{uv}^f \geq -d_{uv}^0 - \sum_{f' \in F, f' \neq f} x_{uv}^{f'}, \quad \forall (u, v) \in A, \\
& \quad \sum_{(u, v) \in A} -c_{uv}^f x_{uv}^f \geq -b_f, \\
& \quad x_{uv}^f \geq 0, \quad \forall (u, v) \in A, \\
& \quad y_u^f \geq 0, \quad \forall u \in V.
\end{align*}
\]

(23)

As observed earlier in Section 2.1.1, the \( y_f \) variables are essentially free variables. In order to simplify analysis, we restrict these variables to be non-negative while ignoring the bounds added in the formulation (10). As we shall see later, it is possible to construct a solution to (23) given a solution to (10). When the interdiction decisions of the agents \( f' \neq f \) are fixed, agent \( f \)'s optimization problem (23) is a linear program (LP). In this case, the KKT conditions are both necessary and sufficient for a given feasible solution to be optimal.

We introduce the following notation to present the KKT conditions for the LP (23) compactly. 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 vectors or matrices of all zeros, of appropriate dimensions, respectively. The objective coefficients for the LP (23), denoted by \( o_f \in \mathbb{R}^{m+n} \) can be given as follows:

\[
o_f = \begin{bmatrix} 0_m \\ \nu_f \end{bmatrix}, \quad \text{where} \quad \nu_f = \begin{cases} 1 & \text{if } u = s_f \\ 0 & \text{if } u \neq s_f, t_f \\ -1 & \text{if } u = t_f \end{cases}.
\]

The right hand sides for the constraints in (23) are denoted using the vector \( r_f(x^{-f}) \in \mathbb{R}^{m+1} \):

\[
r_f(x^{-f}) = \begin{bmatrix} -d_0^f \\ -b_f \end{bmatrix} - \sum_{f' \in F, f' \neq f} \begin{bmatrix} I_m & 0_{m \times n} \\ 0_m^T & 0_n^T \end{bmatrix} \begin{bmatrix} x_{f'}^f \\ y_{f'}^f \end{bmatrix}.
\]

The constraint matrix itself, denoted as \( A_f \in \mathbb{R}^{(m+1) \times (m+n)} \), is

\[
A_f = \begin{bmatrix} I_m & G \\ -c^T & 0_n^T \end{bmatrix}.
\]

Using this notation, the LP (23) can be restated as follows:

\[
\begin{align*}
\text{minimize} & \quad o_f^T \begin{bmatrix} x_f \\ y_f \end{bmatrix} \\
\text{subject to} & \quad A_f \begin{bmatrix} x_f \\ y_f \end{bmatrix} \geq r_f(x^{-f}), \\
& \quad \begin{bmatrix} x_f \\ y_f \end{bmatrix} \geq 0.
\end{align*}
\]

(24)
Let the dual variables for the LP (25) 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 (24) are given by the following system.

\[
\begin{bmatrix}
  x^f \\
  y^f
\end{bmatrix} \leq \begin{bmatrix}
  A^f \\
  \beta^f
\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}
  x^f \\
  y^f
\end{bmatrix}^T \begin{bmatrix}
  \lambda^f \\
  \beta^f
\end{bmatrix} \geq 0,
\]

\[
0 \leq \begin{bmatrix}
  x^f \\
  y^f
\end{bmatrix} \quad \text{and} \quad v^f \geq 0,
\]

\[
o^f - A^f \begin{bmatrix}
  \lambda^f \\
  \beta^f
\end{bmatrix} - v^f = 0.
\]

The KKT system (25) can be rewritten in the following form:

\[
v^f = o^f - A^f \begin{bmatrix}
  \lambda^f \\
  \beta^f
\end{bmatrix} \geq 0, \quad \begin{bmatrix}
  x^f \\
  y^f
\end{bmatrix} \geq 0, \quad \begin{bmatrix}
  x^f \\
  y^f
\end{bmatrix}^T v^f = 0,
\]

\[
t^f = -r^f (x^f) + A^f \begin{bmatrix}
  x^f \\
  y^f
\end{bmatrix} \geq 0, \quad \begin{bmatrix}
  \lambda^f \\
  \beta^f
\end{bmatrix} \geq 0, \quad t^f \begin{bmatrix}
  \lambda^f \\
  \beta^f
\end{bmatrix} = 0.
\]

In this form, it is easy to recognize that for a fixed value of \(x^f\), the KKT system is equivalent to the LCP\((q^f(x^f), M^f)\) where

\[
q^f (x^f) = \begin{bmatrix}
  -r^f (x^f) \\
  -r^f (x^f)
\end{bmatrix} \quad \text{and} \quad M^f = \begin{bmatrix}
  0_{(m+n)\times(m+n)} & -A^f \begin{bmatrix}
  \lambda^f \\
  \beta^f
\end{bmatrix} \\
  A^f & \begin{bmatrix}
  0_{(m+1)\times(m+1)}
\end{bmatrix}
\end{bmatrix}.
\]

The decision variable vector for the LCP is the vector of combined decision variables

\[
w^f = \begin{bmatrix}
  x^f \\
  y^f
\end{bmatrix}.
\]

Each agent’s KKT system (26) is parametrized by the collective decisions of other agents. As mentioned earlier, the optimization problem (23) is completely equivalent to the KKT system (26). In other words, given \((x^f, y^f)\), an agent’s decisions \((x^f, y^f)\) is optimal if and only if it satisfies the system (26). Using this fact, it is straightforward to show that a candidate point \((x^1, x^2, \ldots, x^F)\), where \(x^f = (x^f, y^f)\), is an equilibrium to the DSPI game where each agent solves (23) if and only if it solves the KKT systems (26) for each player \(f \in \mathcal{F}\). As a consequence, the equilibrium problem for the DSPI game under consideration is equivalent to the complementarity problem obtained by stacking the \(F\) systems of (26) for \(f \in \mathcal{F}\). In this case the decision variable is the combined set of primal and dual variables for each agent, denoted by \((w^1, w^2, \ldots, w^F)\).

With some algebraic manipulation, it can be shown that the complementarity system obtained by stacking the \(F\) KKT systems is itself an LCP. Consider the following system obtained from (26) by expanding \(r^f (x^f)\).

\[
v^f = o^f - A^f \begin{bmatrix}
  \lambda^f \\
  \beta^f
\end{bmatrix} \geq 0, \quad \begin{bmatrix}
  x^f \\
  y^f
\end{bmatrix} \geq 0, \quad \begin{bmatrix}
  x^f \\
  y^f
\end{bmatrix}^T v^f = 0,
\]

\[
t^f = \begin{bmatrix}
  q^0 \\
  b^f
\end{bmatrix} + \sum_{f \in \mathcal{F}} \begin{bmatrix}
  \sum_{f \neq j} \begin{bmatrix}
    I_m \\
    0_m
\end{bmatrix} & \begin{bmatrix}
  0_{m \times n} & O_{m \times n}
\end{bmatrix}
\end{bmatrix} \begin{bmatrix}
  x^f \\
  y^f
\end{bmatrix} \geq 0, \quad \begin{bmatrix}
  \lambda^f \\
  \beta^f
\end{bmatrix} \geq 0, \quad t^f \begin{bmatrix}
  \lambda^f \\
  \beta^f
\end{bmatrix} = 0.
\]

15
The interactions between the agent $f$’s decision variables $(x^f, y^f)$ and the KKT system of any other agent $f' \neq f$ can be represented using the matrix $\bar{M}^f$ given below.

\[
\bar{M}^f = \begin{bmatrix}
0_{m \times m} & 0_{m \times n} & 0_{m \times m} & 0_{m \times 1} \\
0_{n \times m} & 0_{n \times n} & 0_{n \times m} & 0_{n \times 1} \\
I_m & 0_{m \times n} & 0_{m \times m} & 0_{m \times 1} \\
0_{1 \times m} & 0_{1 \times n} & 0_{1 \times m} & 0
\end{bmatrix}.
\] (30)

Using this notation, the stacked KKT systems (29) for agents $f = 1, \ldots, F$ can be formulated as LCP$(q, M)$. Here, the vector $q$ is given by

\[
q = \begin{bmatrix}
\bar{q}^1 \\
\bar{q}^2 \\
\vdots \\
\bar{q}^F
\end{bmatrix}, \quad \text{where} \quad \bar{q}^f = \begin{bmatrix}
o^f \\
d^0 \\
b^f
\end{bmatrix},
\] (31)

and the matrix $M$ is given by

\[
M = \begin{bmatrix}
M^1 & M^2 & M^3 & \cdots & M^F \\
M^1 & M^2 & M^3 & \cdots & M^F \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
M^1 & M^2 & \cdots & M^{F-1} & M^F
\end{bmatrix}.
\] (32)

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 DSPI game, as defined in (31) and (32), possesses two properties that allow us to use Lemke’s pivotal algorithm: (i) the matrix $M$ is a copositive matrix, and (ii) $q \in (\text{SOL}(0, M))^\ast$.

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 8.** Let the vector $q$ and the matrix $M$ be as defined in (31) and (32) respectively. Then the matrix $M$ is copositive.

**Proof.** Let $w \in \mathbb{R}^{2m+n+1}_+$. Using the block structure of $M$ given in (32), $w^T M w$ can be decomposed as follows.

\[
w^T M w = \sum_{f=1}^F w^T M^f w^f + \sum_{f=1}^F \sum_{f' \neq f} w^T M^f M^f' w^f'.
\] (33)

We analyze the terms under 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 $\delta^f$.

\[
w^T M^f w^f = \begin{bmatrix}
\chi^f \\
\delta^f
\end{bmatrix} \begin{bmatrix}
0 & -A^T \\
A^T & 0
\end{bmatrix} \begin{bmatrix}
\chi^f \\
\delta^f
\end{bmatrix}
= -\chi^T A^T \delta^f + \delta^T A^f \chi^f
= 0.
\] (34)

16
Now consider any term of the form $w^T \tilde{M} w'$. 

$$
\begin{align*}
    w^T \tilde{M} w' &= \left[ \begin{array}{cccc}
        x^T & y^T & \lambda^T & \beta^T \\
    \end{array} \right] \\
    &= \left[ \begin{array}{c}
        0 \\
        0 \\
        0 \\
        0 \\
    \end{array} \right] \\
    &= \lambda^T x'.
\end{align*}
$$

Combining (34) and (35) we obtain

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

Clearly $w^T \tilde{M} w' \geq 0$, if $w^f \geq 0$ and $w'^f \geq 0$. Thus $w \geq 0$ implies that $w^T M w \geq 0$. \qed

We now show condition (ii), that $q^T w \geq 0 \forall w \in \text{SOL}(0, M)$.

**Lemma 9.** Let the vector $q$ and the matrix $M$ be as defined in (31) and (32) respectively. Then $q \in (\text{SOL}(0, M))^*$. 

**Proof.** Consider any $w \in \text{SOL}(0, M)$. Clearly then, $w^f$ must solve the system (29) for $f = 1, \ldots, F$, with $o^f$, $d^0$ and $b^f$ taking zero values. In this case, considering the primal feasibility of $w^f$ to this system, we obtain the following.

$$
\begin{align*}
    \sum_{a \in A} c^f_{a} x^f_{a} & \leq 0 \\
    y^f_{a} - y^f_{v} + \sum_{f=1}^{F} x^f_{u,v} & \geq 0 \forall (u, v) \in A \ 	ext{for } f = 1, \ldots, F.
\end{align*}
$$

Recall that $c^f_{a} \geq 0$ for all $a \in A$ and $f = 1, \ldots, F$ by assumption. Therefore, (37) implies that $x^f = 0$ for any player $f$. It is easy to see that in this case, we must have

$$
    y^f_{a} - y^f_{v} \geq 0 \forall (u, v) \in A, \ 	ext{for } f = 1, \ldots, F.
$$

The inner-product $q^T w$ can be decomposed as follows.

$$
\begin{align*}
    q^T w &= \sum_{f=1}^{F} q^T w^f \\
    &= \sum_{f=1}^{F} \left( o^f \left[ x^f \right] + d^f \lambda^f + b^f \beta^f \right) \\
    &= \sum_{f=1}^{F} (y^f_{s} - y^f_{t}) + d^f \lambda^f + b^f \beta^f.
\end{align*}
$$
Since \( w \in (\text{SOL}(0, M)) \), \( \lambda^f, \beta^f \geq 0 \). Furthermore, \( d^0, b^f \geq 0 \) by assumption. Clearly, \( d^0 \lambda^f + b^f \beta^f \geq 0 \) for \( f = 1, \ldots, F \).

Thus it only remains to verify that the objective function terms are non-negative. Consider any \( s^f - t^f \) path \( P^f \). By assumption, there must be at least one such path for each player \( f \). By summing up the inequalities (38) over the arcs in the path \( P^f \), we obtain the desired result. In other words,
\[
\sum_{(u,v) \in P^f} y^f_u - y^f_v = y^f_{s^f} - y^f_{t^f} \geq 0. \tag{40}
\]
We have thus shown that \( \bar{q}^T w^f \geq 0 \) for \( f = 1, \ldots, F \). Summing up over the players, the proof is completed.

Using the Lemmas 8 and 9, we can now state the following result about Lemke’s method as it applies to \( \text{LCP}(q, M) \).

**Theorem 10 ([14]).** If \( M \) is copositive and \( q \in (\text{SOL}(q, M))^* \), then Lemke’s method will always compute a solution, if the problem is nondegenerate.

In contrast to the LCP approach for solving DSPI games under continuous interdiction, we have not found a simple or efficient reformulation to solve DSPI games under discrete interdiction (in a centralized manner). In this context, we note that the two-player DSPI game with discrete interdiction may be formulated as a bimatrix game. It is well known that the mixed strategy Nash equilibria to a bimatrix game can be found by solving an LCP. However, for general DSPI games with more than two players, there is no equivalent bimatrix-like game formulation. An integer programming approach to solve the DSPI game under discrete interdiction is provided by the potential function maximization problem used in the proof of Proposition 7. However, the feasible set \( \bar{X} \) is implicitly defined. Constructing a polyhedral representation of such a set poses severe challenges. Motivated by these difficulties, we also explore decentralized approaches to solve DSPI games, which we describe in the next section.

### 4.2 Gauss-Seidel Algorithm

We first present the simplest form of a best-response-based algorithm. The idea is simple: starting with a particular feasible decision variable vector for each agent \( \chi_0 = (\chi^1_0, \chi^2_0, \ldots, \chi^F_0) \), 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 \( \chi^1_0 \) replaced by \( \chi^1_* \). Such an approach is often referred to as a diagonalization scheme or the 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 (12). We will refer to agent \( f \)’s individual’s problem as \( P(\chi^{-f}) \). The Gauss-Seidel iterative procedure is presented in Algorithm 1 below.

The Gauss-Seidel algorithm can be directly applied to compute an equilibrium of a DSPI game with discrete interdiction. For finite termination, we fix a tolerance parameter \( \epsilon \) and use the following stopping criterion:

\[
\|\chi_k - \chi_{k-1}\| \leq \epsilon. \tag{41}
\]

Since the variables \( \chi \) are integral for discrete interdiction problems, choosing \( \epsilon < 1 \) will ensure that the algorithm terminates only when successive outer iterates are equal.

---

3 A detailed discussion of degeneracy and cycling in Lemke’s method can be found in Section 4.9 of [14].
Algorithm 1 Gauss-Seidel Algorithm for a GNEP

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

Step 1:
\[\textbf{for } f = 1, 2, \ldots, F \textbf{ do}\]
\[\quad \text{Set } \chi_{-f}^{k,f} \leftarrow (\chi_{k+1}^1, \ldots, \chi_{k+1}^{f-1}, \chi_k^{f+1}, \ldots, \chi_k^F);\]
\[\quad \text{Solve } P(\chi_{-f}^{k,f}) \text{ to obtain } \chi_f^{k+1}.\]
\[\quad \textbf{end for}\]
\[\text{Set } \chi_{k+1} \leftarrow (\chi_{k+1}^1, \ldots, \chi_{k+1}^F).\]
\[\text{Set } k \leftarrow k + 1.\]

Step 2:
\[\textbf{if } \chi_k \text{ satisfies termination criteria, then STOP.}\]
\[\textbf{else GOTO Step 1.}\]

Proposition 11. Suppose that the Gauss-Seidel algorithm (Algorithm 1), as applied to the DSPI game with discrete interdiction, terminates with $\chi_k$. Then $\chi_k$ is an equilibrium to this problem.

Proof. Since Algorithm 1 terminates with $\chi_k$, it satisfies (41), which implies that $\chi_{k-1} = \chi_k$. 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
\[\chi_f^k = \arg\min_{\chi_f \in \Xi_f(\chi_{-f}^k)} \theta_f(\chi_f^k, \chi_{-f}^k).\]
Clearly, $\chi_k$ must then be an equilibrium.

Proposition 11 establishes Algorithm 1 as a heuristic to solve the DSPI game with discrete interdiction. However, there is no guarantee that the algorithm will in fact converge, despite the fact that the DSPI game possesses a potential function. While general results on convergence of best response dynamics for potential games have been well-established in literature, the difficulty here in showing convergence lies in the fact that we are dealing with a GNEP, in which each agent’s feasible region is affected by other agents’ actions, and such coupling constraints do not have the same functional form for each agent. As a result, any intermediate points resulting from an agent’s best responses need not be feasible in the other agents’ problems.

We note however that it is possible to detect when the algorithm fails to converge. Recall that $\Xi_f(\chi_{-f}^k) \subseteq K_f^f$ for each agent $f \in F$, where $K_f^f$ is defined in (17). Moreover, the set $\prod_{f=1}^F K_f^f$ is finite. Any intermediate point $\chi_k$ generated by Algorithm 1 must certainly satisfy the budgetary constraints on $x_k^f$ and the bound constraints on $y_k^f$ for each agent $f$. Therefore $\chi_k \in \prod_{f=1}^F K_f^f$. In other words, the set of possible points $\chi_k$ generated by Algorithm 1 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 might fail to converge.

For DSPI games with continuous interdiction, establishing the convergence result for a best-response type algorithm is more involved. Even for typical Nash equilibrium problems with no constraint interactions, the simple implementation of the Gauss-Seidel algorithm described in Algorithm 1 may not work. To obtain better convergence properties, we need a regularization scheme, as shown next.

4.3 Regularized Gauss-Seidel Algorithm

It can be shown that the basic diagonalization scheme of solving individual agent problems in sequence and updating agent decision variables at each step may not converge to an equilibrium even for GNEPs with
favorable properties such as continuously differentiable and convex objective functions. However, Facchinei et al. [25] 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

\[
\text{maximize } \theta^f(\chi^f, \chi^{-f}) - \tau \| \chi^f - \bar{\chi}^f \|^2 \\
\text{subject to } \chi^f \in \Xi^f(\chi^{-f}),
\]

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 (42) is solved in a regularized Gauss-Seidel iteration. For ease of notation, we will refer to problem (42) as \( R(\chi^{-f}, \bar{\chi}^f) \).

The regularized Gauss-Seidel procedure is given in Algorithm 2. As the name indicates, the agents’ optimization problems contain a regularization term, and are solved in sequence. The regularization term ensures that problem (42) has a unique optimal solution so that the algorithm is well defined. The algorithm is set to terminate when the outer iterates become sufficiently close to each other; in other words, the termination condition is given in (41).

Algorithm 2 Regularized Gauss-Seidel Algorithm for a GNEP

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

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

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

Set \( \bar{\chi}^f \leftarrow \chi_f^k \); Solve \( R(\chi_{k+1}^f, \bar{\chi}^f) \) to obtain \( \chi_{k+1}^f \).

end for

Set \( \chi_{k+1} \leftarrow (\chi_{k+1}^1, \ldots, \chi_{k+1}^F) \).

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

Step 2:
if \( \chi_k \) satisfies termination criteria, then STOP.
else GOTO Step 1.

This version of the algorithm was originally presented in [25] to solve potential GNEPs with shared constraints. Since DSPI games are GNEPs of non-shared constraints, we use Algorithm 2 as a heuristic algorithm to solve DSPI games under continuous interdiction.

Proposition 12. Let \( \{\chi_k\} \) be the sequence generated by applying Algorithm 2 to the DSPI problem under continuous interdiction, wherein each agent solves (10). Suppose \( \{\chi_k\} \) converges to \( \bar{\chi} \). Then \( \bar{\chi} \) is an equilibrium to the DSPI problem.

The proof of this theorem, though a straightforward adaptation of Theorem 4.3 in Facchinei et al. [25] does differ in one key aspect. We 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 \( \chi_{k,f} \) converge to \( \bar{\chi} \), a fact key to proving that \( \bar{\chi} \) is indeed an equilibrium. In contrast, for GNEPs with shared constraints, the feasibility of the intermediate points ensures their convergence even to cluster points of the sequence generated by the algorithm. The complete proof of Proposition 12 is given in Appendix A.
4.4 Numerical Results

We use the algorithms presented in the previous section to study several instances of DSPI games. The decentralized algorithms were implemented in MATLAB R2010a with CPLEX v12.2 as the optimization solver. The LCP formulation for the DSPI game with continuous interdiction was solved using the MATLAB interface for the complementarity solver PATH [26]. 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.

For DSPI games with discrete interdiction, we used Algorithm 1. For DSPI games with continuous interdiction, we applied a combination of Algorithm 1 and Algorithm 2. In particular, we first tried Algorithm 1 and then used Algorithm 2 if Algorithm 1 failed to converge to an equilibrium. We pursued this strategy since the number of outer iterations until Algorithm 2 converged was found to be quite sensitive to the regularization parameter $\tau$, typically resulting in slow convergence rates for the algorithm. Since the running time for Algorithm 1 especially with the outer iterations restricted to a maximum of 1000, was quite short relative to the running time for Algorithm 2, it seemed reasonable to try using Algorithm 1 first, and use Algorithm 2 only as necessary.

Computing Equilibria

First, we applied the algorithm to Example 2 in Section 3.3.1 which is a DSPI game with continuous interdiction. In particular, the network is given in Figure 1 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 1 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 1. The column $x_0$ represents the starting interdiction vector for each agent; while $x_N^1$ 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 1 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 1, $p_1$ and $p_2$, give the shortest path lengths for agents 1 and 2 respectively, at the equilibrium $\chi_N$.

| $x_0$          | $x_N^1$               | $x_N^2$               | $p_1$       | $p_2$       |
|----------------|-----------------------|-----------------------|-------------|-------------|
| (0, 0, 0, 0)   | (0, 0, 0.5, 0, 0, 0)  | (0, 0, 0.1667, 0.667, 0) | 0.6667      | 0.6667      |
| (0, 0, 2, 0)   | (0, 0, 0.6, 0.4, 0)   | (0, 0, 0.0667, 0.667, 0) | 0.6667      | 0.6667      |
| (0, 0, 0, 0)   | (0, 0, 0.4, 0.6, 0)   | (0, 0, 0.2667, 0.667, 0) | 0.6667      | 0.6667      |
| (0, 0, 0, 0)   | (0, 0, 0.35, 0.65, 0) | (0, 0, 0.3167, 0.667, 0) | 0.6667      | 0.6667      |
| (0, 0, 0, 0)   | (0, 0, 0.65, 0.35, 0) | (0, 0, 0.0167, 0.667, 0) | 0.6667      | 0.6667      |
| (0.25, 0.25)   | (0, 0, 0.625, 0.375)  | (0, 0, 0.0417, 0.2917, 0) | 0.6667      | 0.6667      |
| (0, 0, 0, 0)   | (0, 0, 0.375, 0.625)  | (0, 0, 0.2917, 0.2917, 0) | 0.6667      | 0.6667      |
| (0, 0, 0, 0)   | (0, 0, 0.425, 0.575)  | (0, 0, 0.2417, 0.0917, 0) | 0.6667      | 0.6667      |
| (0.15, 0.15)   | (0, 0, 0.375, 0.425)  | (0, 0, 0.0917, 0.2417, 0) | 0.6667      | 0.6667      |

Example 4. To test the algorithm on problems with larger scale, we expanded the network in Example 2 with varying graph sizes and numbers of agents. For $F$ agents, the graph contains $2(F + 1)$ vertices with the edges as shown in Figure 2. 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 players and are given as the arc labels in Figure 2. All the agents have an interdiction budget of 1. The cost parameter $\epsilon$ is chosen as 2. For discrete interdiction on these graphs, arc extensions are assumed to be by a length of 1.

The running time and iterations required to compute equilibria for these instances are summarized in Table 2 and Table 3. Table 2 gives the number of outer iterations and runtime for Algorithm 1 over these instances with continuous interdiction. For an empirical comparison between the decentralized and centralized approaches, the performance of Lemke’s method for the LCP formulation is given in the last column of the table. The results indicate that the running time for the centralized 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 Algorithm 1 plays an important role. Indeed it was found that the algorithm could fail to converge for certain agent orders, while succeeding to find equilibria quickly for the same instance with a different ordering of agents. 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 Algorithm 1 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 Algorithm 1, the regularized method Algorithm 2 converged to a GNE within 394 outer-iterations with a runtime of 28 wall-clock seconds.

**Computation of Efficiency Losses**

Using the decentralized algorithm and its potential to find multiple equilibria by starting at different points, we empirically study the efficiency loss of decentralized interdiction strategies in DSPI games. We focus first on Example 4 with the underlying network represented in Figure 2. Before computing the average efficiency losses empirically using our algorithms, we first establish a theoretical bound on the worst-case price of anarchy, for the purpose of comparison.

Recall that there are $F$ agents and the source-target pair for agent $f$ is $(a_1, b_{f+1})$. Noting that all paths for all agents contain either the arc $(a_1, a_2)$ or the arc $(a_1, b_1)$, 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 $n/(2 + \epsilon)$, giving a shortest path length of $n/(2 + \epsilon)$ 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 $(n + \epsilon/2)/(2 + \epsilon)$.
Table 2: Number of iterations and running times for DSPI Example 3 under continuous interdiction.

| # Agents | # Iterations | Runtime (s) | LCP  |
|----------|--------------|-------------|------|
| 5        | 3            | 0.0205      | 0.0290 |
| 10       | 5            | 0.0290      | 0.1833 |
| 15       | 11           | 0.1103      | 0.7534 |
| 20       | 5            | 0.0723      | 2.1106 |
| 25       | 13           | 0.2609      | 4.8167 |
| 30       | 15           | 0.4070      | 10.2256 |
| 35       | 10           | 0.3605      | 17.7387 |
| 40       | 41           | 1.7485      | 30.2382 |
| 45       | 12           | 0.6601      | 48.6280 |
| 50       | 12           | 0.7981      | 75.0420 |

Table 3: Number of iterations and running times for DSPI Example 3 under discrete interdiction.

| # Agents | # Iterations | Runtime (s) |
|----------|--------------|-------------|
| 5        | 5            | 0.1776      |
| 10       | 3            | 0.1627      |
| 15       | 3            | 0.2419      |
| 20       | 3            | 0.3164      |
| 25       | 3            | 0.4005      |
| 30       | 3            | 0.5155      |
| 35       | 3            | 0.5948      |
| 40       | 3            | 0.7387      |
| 45       | 3            | 0.8794      |
| 50       | 3            | 1.0385      |

A Nash equilibrium to this problem 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 $n/(n+1)$. 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 arc 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 $1 + \epsilon$. 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 $n/(2 + \epsilon)$ for each agent, and a Nash equilibrium that has an objective value of $n/(n + 1)$ for each agent. Therefore, the worst-case price of anarchy for the DSPI game depicted in Figure 2 must be at least $(n + 1)/(2 + \epsilon)$.

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 purposes of comparison, the numerical results are plotted in Figure 3 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 the computational framework we presented.

**Example 5.** We further tested the decentralized algorithms for continuous interdiction on random graphs to study average efficiency losses of equilibria of DSPI games on networks with different topologies.
generate random graphs, we took the number of vertices and the density of the graph – the number of arcs in the random graph divided by the maximum possible number of arcs – as inputs. The number of agents was chosen randomly from the interval \((0, |V|/2)\); one agent set size was 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 players until the desired density was reached. We thus ensured connectivity between the source-target pairs for each player. 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 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, agent set size 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 4. 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 in the density of the underlying network.

5 Conclusions and Future Work

In this work, we introduced decentralized network interdiction (DNI) games and gave formulations for one such class of games – decentralized shortest path interdiction (DSPI) games. We analyzed the theoretical properties of DSPI games: in particular, we gave conditions for the existence of equilibria and examples where multiple equilibria exist. We also showed that DSPI games belong to a special class of games called potential games. This property was key in establishing several of the theoretical results we presented.
We showed that the DSPI game under continuous interdiction is equivalent to a linear complementarity problem, which can be solved by the Lemke’s algorithm. This constitutes a convergent centralized method to solve such problems. We also presented decentralized heuristic algorithms to solve DSPI games under both continuous and discrete interdiction. Finally, we used these algorithms to empirically evaluate the worst case and average efficiency loss of DSPI 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 DSPI games, we also made the assumption in this paper that the games have complete information; that is, the normal form of the game – the set of agents, agents’ feasible action spaces, and their utility 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 under grant FA9550-12-1-0275.

References

[1] S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty. On Nash equilibria for a network creation game. In Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 89–98. ACM, 2006.
[2] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[3] E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler. Near-optimal network design with selfish agents. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 511–520. ACM, 2003.

[4] N. Assimakopoulos. A network interdiction model for hospital infection control. *Computers in Biology and Medicine*, 17(6):413–422, 1987.

[5] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 57–66. ACM, 2005.

[6] M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali. *Linear Programming and Network Flows*. Wiley, 2011.

[7] M.J. Beckmann, C.B. McGuire, and C.B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.

[8] G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *Algorithms–ESA 2005*, pages 59–70. Springer, 2005.

[9] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 67–73. ACM, 2005.

[10] R.L. Church, M.P. Scaparra, and R.S. Middleton. Identifying critical infrastructure: The median and covering facility interdiction problems. *Annals of the Association of American Geographers*, 94(3):491–502, 2004.

[11] R. Cole, Y. Dodis, and T. Roughgarden. How much can taxes help selfish routing? *Journal of Computer and System Sciences*, 72(3):444–467, 2006.

[12] J. Corbo and D. Parkes. The price of selfish behavior in bilateral network formation. In *Proceedings of the 24th Annual ACM Symposium on Principles of Distributed Computing*, pages 99–107. ACM, 2005.

[13] J.R. Correa, A.S. Schulz, and N.E. Stier-Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004.

[14] Richard W Cottle, Jong-Shi Pang, and Richard E Stone. *The linear complementarity problem*, volume 60. Siam, 2009.

[15] G. Debreu. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences of the United States of America*, 38(10):886, 1952.

[16] E.D. Demaine, M. Hajiaghayi, H. Mahini, and M. Zadimoghaddam. The price of anarchy in network creation games. In *Proceedings of the 26th Annual ACM Symposium on Principles of Distributed Computing*, pages 292–298. ACM, 2007.

[17] A. Dreves, F. Facchinei, C. Kanzow, and S. Sagratella. On the solution of the KKT conditions of generalized Nash equilibrium problems. *SIAM Journal on Optimization*, 21(3):1082–1108, 2011.

[18] A. Dreves, C. Kanzow, and O. Stein. Nonsmooth optimization reformulations of player convex generalized Nash equilibrium problems. *Journal of Global Optimization*, 53(4):587–614, 2012.
[19] A. Dreves, A. von Heusinger, C. Kanzow, and M. Fukushima. A globalized Newton method for the computation of normalized Nash equilibria. *Journal of Global Optimization*, 56(2):327–340, 2013.

[20] A. Fabrikant, A. Luthra, E. Maneva, C.H. Papadimitriou, and S. Shenker. On a network creation game. In *Proceedings of the 22nd Annual ACM Symposium on Principles of Distributed Computing*, pages 347–351. ACM, 2003.

[21] F. Facchinei, A. Fischer, and V. Piccialli. On generalized Nash games and variational inequalities. *Operations Research Letters*, 35(2):159 – 164, 2007.

[22] F. Facchinei and C. Kanzow. Generalized Nash equilibrium problems. *4OR*, 5(3):173–210, 2007.

[23] F. Facchinei and C. Kanzow. Penalty methods for the solution of generalized Nash equilibrium problems. *SIAM Journal on Optimization*, 20(5):2228–2253, 2010.

[24] F. Facchinei and J. S. Pang. Exact penalty functions for generalized Nash problems. In G. Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and Its Applications*, pages 115–126. Springer US, 2006.

[25] F. Facchinei, V. Piccialli, and M. Sciandrone. Decomposition algorithms for generalized potential games. *Computational Optimization and Applications*, 50:237–262, 2011.

[26] Michael C Ferris and Todd S Munson. Interfaces to path 3.0: Design, implementation and usage. *Computational Optimization and Applications*, 12(1-3):207–227, 1999.

[27] L. Fleischer, K. Jain, and M. Mahdian. Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 277–285. IEEE, 2004.

[28] M. Fukushima. Restricted generalized Nash equilibria and controlled penalty algorithm. *Computational Management Science*, 8(3):201–218, 2011.

[29] P.M. Ghare, D.C. Montgomery, and W.C. Turner. Optimal interdiction policy for a flow network. *Naval Research Logistics Quarterly*, 18(1):37–45, 1971.

[30] P.T. Harker. Generalized Nash games and quasi-variational inequalities. *European Journal of Operational Research*, 54(1):81 – 94, 1991.

[31] T. Ichiishi. *Game Theory for Economic Analysis*. Academic Press New York, 1983.

[32] E. Israeli and R.K. Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002.

[33] R. Johari and J.N. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3):407–435, 2004.

[34] R. Johari and J.N. Tsitsiklis. Efficiency of scalar-parameterized mechanisms. *Operations Research*, 57(4):823–839, 2009.

[35] B.B.H. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Algorithms and combinatorics. Springer-Verlag Berlin Heidelberg, 2012.

[36] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 404–413. Springer, 1999.
[37] J.B. Krawczyk and S. Uryasev. Relaxation algorithms to find Nash equilibria with economic applications. *Environmental Modeling & Assessment*, 5(1):63–73, 2000.

[38] D. Monderer and L.S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.

[39] D.P. Morton, F. Pan, and K.J. Saeger. Models for nuclear smuggling interdiction. *IIE Transactions*, 39(1):3–14, 2007.

[40] R.B. Myerson. Refinements of the Nash equilibrium concept. *International Journal of Game Theory*, 7(2):73–80, 1978.

[41] K. Nabetani, P. Tseng, and M. Fukushima. Parametrized variational inequality approaches to generalized Nash equilibrium problems with shared constraints. *Computational Optimization and Applications*, 48(3):423–452, 2011.

[42] J.S. Pang and M. Fukushima. Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games. *Computational Management Science*, 2(1):21–56, 2005.

[43] A.C. Pigou. *The Economics of Welfare*. Transaction Publishers, 1924.

[44] R.T. Rockafellar and R. Wets. *Variational Analysis: Grundlehren Der Mathematischen Wissenschaften*, volume 317. Springer Verlag, 1998.

[45] J.B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica: Journal of the Econometric Society*, pages 520–534, 1965.

[46] R.W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[47] T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.

[48] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.

[49] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley Series in Discrete Mathematics & Optimization. John Wiley & Sons, 1998.

[50] J. C. Smith and C. Lim. Algorithms for network interdiction and fortification games. In A. Chinchuluun, P.M. Pardalos, A. Migdalas, and L. Pitsoulis, editors, *Pareto Optimality, Game Theory And Equilibria*, volume 17 of *Springer Optimization and Its Applications*, pages 609–644. Springer New York, 2008.

[51] S. Suri, C.D. Tóth, and Y. Zhou. Selfish load balancing and atomic congestion games. *Algorithmica*, 47(1):79–96, 2007.

[52] D.R.M. Thompson and K. Leyton-Brown. Computational analysis of perfect-information position auctions. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, pages 51–60. ACM, 2009.

[53] S. Uryasev and R.Y. Rubinstein. On relaxation algorithms in computation of noncooperative equilibria. *IEEE Transactions on Automatic Control*, 39(6):1263–1267, 1994.
A. Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, pages 416–425. IEEE, 2002.

A. von Heusinger and C. Kanzow. SC1 optimization reformulations of the generalized Nash equilibrium problem. Optimization Methods Software, 23(6):953–973, 2008.

A. von Heusinger and C. Kanzow. Optimization reformulations of the generalized Nash equilibrium problem using Nikaido-Isoda-type functions. Computational Optimization and Applications, 43(3):353–377, 2009.

J.G. Wardrop. Some theoretical aspects of road traffic research. In ICE Proceedings: Engineering Divisions, volume 1, pages 325–362. Thomas Telford, 1952.

R.K. Wood. Deterministic network interdiction. Mathematical and Computer Modelling, 17(2):1–18, 1993.

J. Zhang, B. Qu, and N. Xiu. Some projection-like methods for the generalized Nash equilibria. Computational Optimization and Applications, 45(1):89–109, 2010.

A Proof of Proposition 12

Since \( \chi_k \to \bar{\chi} \) we must have \( \chi^f_k \to \bar{\chi}^f \) and

\[
\lim_{k \to \infty} \left\| \chi^f_{k+1} - \chi^f_k \right\| = 0. \tag{43}
\]

By construction of \( \chi_{k,f} \), (43) implies that

\[
\lim_{k \to \infty} \chi_{k,f} = \bar{\chi}. \tag{44}
\]

By Step 1 of Algorithm 2 \( \chi^f_{k+1} \in \Xi^f(\chi^f_k) \). Since \( \chi^f_{k+1} \to \bar{\chi}^f \), \( \chi^f_k \to \bar{\chi}^f \), and \( \Xi^f(\chi^f_k) \) is defined by linear inequalities parametrized by \( \chi^f_k \), it is straightforward to see by continuity arguments that \( \bar{\chi}^f \in \Xi^f(\bar{\chi}^f) \). In other words, \( \bar{\chi} \) is feasible for every agent’s optimization problem (12).

We claim that for each agent \( f \in \mathcal{F} \)

\[
\theta^f(\bar{\chi}^f, \bar{\chi}^f) \geq \theta^f(\chi^f_k, \bar{\chi}^f), \quad \forall \chi^f_k \in \Xi^f(\bar{\chi}^f). \]

For the purposes of establishing a contradiction, let there be an agent \( \bar{f} \) and a vector \( \xi^f \in \Xi^f(\bar{\chi}^f) \) such that

\[
\theta^f(\bar{\chi}^f, \bar{\chi}^f) < \theta^f(\xi^f, \bar{\chi}^f). \tag{45}
\]

We established in the proof of Proposition 6 that the set valued mapping \( \Xi^f(\cdot) \) satisfies inner semicontinuity relative to its domain. Using the definition of inner semicontinuity (cf. Chapter 5, Section B), and because \( \bar{\chi}^f \in \text{dom}(\Xi^f(\cdot)) \), we then have

\[
\liminf_{\xi^f \rightarrow \bar{\chi}^f} \Xi(\xi^f) \supset \Xi(\bar{\chi}^f), \tag{46}
\]

where the limit in (45) is given by the following:

\[
\liminf_{\xi^f \rightarrow \bar{\chi}^f} \Xi(\xi^f) = \left\{ u^f \mid \forall \chi^f_k \to \bar{\chi}^f, \exists u^f_k \to u \text{ with } u^f_k \in \Xi^f(\chi^f_k) \right\}. \tag{46}
\]
Since $\xi^f \in \Xi(\bar{x} - f)$, equations (44), (45) and (46) allow us to construct a sequence $\xi_k^f \in \Xi(\chi - f)$ such that $\xi_k^f \to \xi^f$ as $k \to \infty$.

Let $d^f = (\xi^f - \bar{x}^f)$. Then by the subdifferentiality inequality for concave functions we must have

$$\theta^f(\bar{x}^f, \bar{x}^f - d^f) > 0.$$  \hspace{1cm} (47)

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

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

We then have

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

Note that $\chi_{k+1}^f$ is obtained by solving the problem $R(\chi_{k,f}, \chi_{k,f})$. In other words, $\chi_{k+1}^f$ maximizes $\Phi^f(\xi_k^f, \chi_{k,f}^f, \chi_{k,f}^f)$ over the set $\Xi^f(\chi_{k,f}^f)$. Since this is a concave maximization problem, we then apply first order optimality conditions to obtain the following.

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

$$+ 2\tau(\chi_{k+1}^f - \chi_{k}^f)(\xi_k^f - \chi_k^f)^T$$

$$\leq 0.$$ \hspace{1cm} (48)

Passing to the limit $k \to \infty$, $k \in K$ and using (44) we obtain

$$0 \geq \theta^f(\bar{x}^f, \bar{x}^f - (\xi_k^f - \bar{x}^f))$$

which contradicts (47).