An Algorithm for Resilient Nash Equilibrium Seeking in the Partial Information Setting

Dian Gadjov and Lacra Pavel, Senior Member, IEEE

Abstract—Current research in distributed Nash equilibrium (NE) seeking in the partial information setting assumes that information is exchanged between agents that are “truthful.” However, in general noncooperative games, agents may consider sending misinformation to neighboring agents with the goal of further reducing their cost. In addition, communication networks are vulnerable to attacks from agents outside the game as well as communication failures. In this article, we propose a distributed NE-seeking algorithm that is robust against adversarial agents that transmit noise, random signals, constant singles, deceitful messages, as well as being resilient to external factors such as dropped communication, jammed signals, and man-in-the-middle attacks. The core issue that makes the problem challenging is that agents have no means of verifying if the information they receive is correct, i.e., there is no “ground truth.” To address this problem, we use an observation graph, which gives truthful action information, in conjunction with a communication graph, which gives (potentially incorrect) information. By filtering information obtained from these two graphs, we show that our algorithm is resilient against adversarial agents and converges to the NE.

Index Terms—Distributed algorithms/control, game theory, nonlinear systems, optimization, sensor networks.

I. INTRODUCTION

DESIGNING distributed Nash equilibrium (NE) seeking algorithms is currently an active research area. This is due to the wide range of problems that can be formulated as a network scenario between self-interested agents. Problems that have been formulated as a distributed NE problem are demand-side management for smart grids [1], electric vehicles [2], competitive markets [3], network congestion control [4], power control and resource sharing in wireless/wired peer-to-peer networks, cognitive radio systems [5], etc.

Classically, NE problems were solved for the full-decision information setting, i.e., where all players have the knowledge of every agent’s decision/action [6]. This requires a centralized system to disseminate the action information [7] or agents must transmit their action information to all the other agents. However, there is a large class of problems where this assumption is unreasonable. For example, either when a central coordinator does not exist or it is infeasible to have a centralized system, e.g., due to cost, computational burden, agent limitations, etc. In recent years, research has focused on the partial-decision information setting, where agents do not know the actions of the other agents, but may communicate with neighboring agents [8], [9], [10]. Agents communicate to learn the true value of the actions of all the agents in the network.

All such existing algorithms make the implicit assumption that agents share information truthfully. For some problems, this assumption is reasonable because the agents are working for a common goal, the agents are guided by a central authority, or there are systems in place so that sending misinformation is impossible. Even in these settings, the agents are susceptible to external factors, such as communication failures and communication attacks. Moreover, in general noncooperative games, agents might have an incentive to not be truthful if deceiving others can minimize their cost further. Motivated by this problem, we propose an NE seeking algorithm that is robust against adversarial agents and communication failures/attacks.

In the consensus problem literature, there are algorithms designed to reach a consensus in settings where there are malicious agents under a wide variety of communication assumptions [11], [12], [13], [14], [15], [16], [17]. However, these algorithms are solely focused on just solving the consensus problem, which is just a component of the NE seeking process. These algorithms do not account for the optimization problem that each agent is trying to solve.

In the distributed optimization literature, ideas from the consensus problem have been applied to design resilient optimization algorithms. However, agents only converge to a convex combination of the local function minimizers and are unable to guarantee convergence to the optimal solution [18], [19], [20], [21]. The reason for this shortcoming is that adversarial agents are indistinguishable from a normal agent with a modified (and valid) cost function [19, Th. 4.4].

In the game theory literature, malicious agents have been studied in various settings [22], [23]. However, the focus of the research is on understanding the behavior of agents or designing mechanisms to remove the incentive to act maliciously. In the distributed NE seeking literature, there are very few results that attempt to tackle this problem. Feng and Hu [24] proposed an NE seeking algorithm against denial-of-service attacks. However, this article makes assumptions on the frequency and duration

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
of attacks, and assumes that agents can detect when a communication link is attacked. While the work in [24] does handle a specific external attack on the network, it does not deal with malicious agents in the game.

Contributions: In this article, we propose a novel distributed NE seeking algorithm that is robust to adversarial agents. The algorithm is resilient to both external factors, such as attacks from agents outside the game and communication faults, and internal adversaries that transmit messages to neighboring agents with the intention of deceiving them to reduce their own cost. To achieve this, we utilize a so-called observation graph, through which actions can be directly observed and are not susceptible to being tampered with. To the best of our knowledge, this is the first such result in the literature. Compared with [25], this article relaxes some of the assumptions, provides proofs for all the results that were absent in [25] due to space limitations, and provides a larger scale example.

The rest of this article is organized as follows. Section I-A gives preliminary background. Section II formulates the problem. In Section III, we introduce the observation graph and discuss properties of graphs. The proposed algorithm is presented in Section IV. The convergence analysis is presented in Section V. Numerical results are provided in Section VI. Finally, Section VII concludes this article.

A. Notation and Terminology

Let \( (1_n)_{0_n} \) denote the vector of all (ones) zeros of dimension \( n \), and \( I_n \) is the \( n \times n \) identity matrix. To ease notation, we will drop the subscript when the dimension can be inferred. Given a set \( C \subset S \), let \( \overline{C} = S \backslash C \) denote the complement of set \( C \). Given an ordered index set \( \mathcal{I} = \{1, 2, \ldots, n\} \), let \( \text{col}(x_i)_{i \in \mathcal{I}} \triangleq [x_{i1}, x_{i2}, \ldots, x_{in}]^T \) and \( \text{row}(x_i)_{i \in \mathcal{I}} \triangleq [x_{1i}, x_{2i}, \ldots, x_{ni}]^T \). Given matrices \( A \) and \( B \), we use \( A \succ B \) \( (A \succeq B) \) to denote that \( A - B \) is positive definite (positive semidefinite). We denote the Euclidean norm of \( x \) as \( \|x\|^2 \triangleq \langle x, x \rangle \). For a given symmetric matrix \( M \succeq 0 \), let \( \langle x, y \rangle_M \triangleq \langle x, My \rangle \) and \( \|x\|_M^2 \triangleq \langle x, x \rangle_M \), which is a norm when \( M \succeq 0 \).

A directed graph is denoted by \( \mathcal{G} = (\mathcal{N}, \mathcal{E}) \), where \( \mathcal{N} \) is the set of nodes and \( \mathcal{E} \subset \mathcal{N} \times \mathcal{N} \) is the set of edges. Let \((j, i) \in \mathcal{E}\) denote an edge from node \( j \) to node \( i \). For graph \( \mathcal{G} \), the set of in-neighbors to node \( i \) is denoted \( \mathcal{N}_{i}^\text{in}(\mathcal{G}) \triangleq \{ j | (j, i) \in \mathcal{E} \} \) and the set of out-neighbors is \( \mathcal{N}_{i}^\text{out}(\mathcal{G}) \triangleq \{ j | (i, j) \in \mathcal{E} \} \). A path from node \( i \) to node \( j \) is a sequence of nodes \( v_1, v_2, \ldots, v_n \) such that \( v_1 = i, v_n = j \), and \( (v_k, v_{k+1}) \in \mathcal{E} \forall k \in \{1, 2, \ldots, n - 1\} \). A graph is rooted at \( i \) if there is a path from \( i \) to every node \( j \in \mathcal{N}_{i}^\text{in}(\mathcal{G}) \). A graph is strongly connected if there is a path from every node to every other node.

II. PROBLEM FORMULATION

Consider a set of \( N \) players/agents denoted \( \mathcal{N} = \{1, 2, \ldots, N\} \). Each player \( i \in \mathcal{N} \) selects an action \( x_i = \text{col}(x_{iq})_{q \in \{1, 2, \ldots, n_i\}} \in \Omega_i \subset \mathbb{R}^{n_i} \), where \( \Omega_i \) is the constraint set and \( x_{iq} \in \mathbb{R} \) is the \( q \)th component of agent \( i \)’s action. Let \( x \triangleq \text{col}(x_i)_{i \in \mathcal{N}} \in \Omega \) denote the action profile of all the players’ actions, where \( \Omega \triangleq \bigcap_{i \in \mathcal{N}} \Omega_i \subset \mathbb{R}^n \) and \( n = \sum_{i \in \mathcal{N}} n_i \). In addition, we will denote \( x = (x_1, x_{-i}) \), where \( x_{-i} \in \Omega_{-i} = \bigcap_{j \neq i} \Omega_j \) is the action of all players except for \( i \). Each agent \( i \) has a cost function \( J_i : \Omega \to \mathbb{R} \), which we denote as \( J_i(x_i, x_{-i}) \) to emphasize that the cost is dependent on \( i \)’s own and the other agents actions.

In a game \( G \), agents have full information (FI) if agent \( i \) knows \( x_{-i} \), and partial information (PI) otherwise. When agents have FI, the problem that each agent tries to solve is

\[
\min_{x_i} \quad J_i(x_i, x_{-i})
\]

s.t. \( x_i \in \Omega_i \). (NE-FI)

Definition 1: Given a game \( G \), an action profile \( x^* = (x_1^*, x_{-1}^*) \) is an NE if \( x^* \) solves (NE-FI) for all agents \( i \in \mathcal{N} \).

We make the following assumption to ensure that the NE of the game exists and is unique [6, Prop. 2.3.3(b)].

Assumption 1: Consider the set \( \Omega = \mathbb{R}^n \). For each player \( i \in \mathcal{N} \), given any \( x_{-i} \), \( J_i(x_i, x_{-i}) \) is continuously differentiable and convex in \( x_i \). The pseudogradient \( F(x) \triangleq \text{col}(\frac{\partial J_i(x)}{\partial x_i})_{i \in \mathcal{N}} \) is \( \mu \)-strongly monotone and \( L \)-Lipschitz.

Remark 1: Under Assumption 1, the NE can be characterized as the action profile \( x^* \), where the pseudogradient is \( 0 \), i.e., \( F(x^*) = 0 \).

For NE problems, it is classically assumed that agents know the other agents’ actions, i.e., agent \( i \) knows \( x_{-i} \) [6]. However, in many scenarios, this assumption is impractical. For example, consider a scenario where drones are ordered to arrange themselves into a certain configuration. It may be unreasonable to assume that every drone can know where every other drone is, since the distances between agents could be quite large.

If agents only have PI, then they cannot solve (NE-FI) because they do not know \( x_{-i} \) fully. One way to try and resolve this issue is to have agents share information to neighboring agents over a communication graph \( \mathcal{G}_c \). In the drone example, this could mean that drones have a transmitter that can only broadcast information over a small radius such that only nearby agents can receive this message.

Definition 2: A graph \( \mathcal{G}_c = (\mathcal{N}, \mathcal{E}_c) \) is a communication graph where the node set \( \mathcal{N} \) is the set of agents and the edge set \( \mathcal{E}_c \) describes the flow of information between agents. An edge \( e_{ij} \in \mathcal{E}_c \) if agent \( j \) gives information to agent \( i \).

In the PI case, agent \( i \) does not know \( x_{-i} \) and uses \( \mathcal{G}_c \) to learn \( x_{-i} \). To do so, agent \( i \) maintains a vector \( x_i^t \triangleq \text{col}(x_{ij}^t)_{j \in \mathcal{N} \backslash \{i\}} \in \mathbb{R}^{n_i} \), where \( x_{ij}^t \in \mathbb{R}^{n_i} \) is agent \( i \)’s genuine estimate/belief of agent \( j \)’s action. Furthermore, \( x_{ij}^t = \text{col}(x_{iq}^t)_{q \in \{1, 2, \ldots, n_i\}} \in \mathbb{R}^{n_i} \) is agent \( j \)’s genuine belief of the \( q \)th component of agent \( j \)’s action. Note that \( x_{ij}^t \) is agent \( i \)’s estimate of their own action, since they know their own action \( x_{ij}^t = x_{ij} \). Ideally, all agents \( i \) would like to use the communication graph to update their beliefs such that \( x_{ij}^t = x_{ij} \), i.e., their belief matches reality. We denote all the agents’ estimates/beliefs stacked into a single vector as \( x^t \triangleq \text{col}(x_{ij}^t)_{i \in \mathcal{N}} \). We denote the communicated message that agent \( j \) sends to agent \( i \), via \( \mathcal{G}_c \), about the action of agent \( m \) as \( y_{ij}^t \in \mathbb{R}^{n_m} \). Furthermore, \( y_{ij}^t = \text{col}(y_{ijq}^t)_{q \in \{1, 2, \ldots, n_m\}} \).

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where \( y_{ij} \in \mathbb{R} \) is agent \( j \)'s communicated message to agent \( i \) about the \( q \)th component of agent \( m \)'s action. In addition, let \( y^i = \text{col}(y_{im})_{m \in \mathcal{N}} \in \mathbb{R}^n \), \( y^i = \text{col}(y_{ij})_{j \in \mathcal{N}} \in \mathbb{R}^n \), and, without loss of generality, \( y^i = x^i \), i.e., the message agent \( i \) "sends to himself" is equal to agent \( i \)'s genuine belief. In general, agent \( i \) will update its estimate/actions and communicate messages via

\[
\begin{align*}
x^i[k + 1] &= f_i \left( x^i[k], y^i[k] \right) \\
y^i[k + 1] &= h_{ji} \left( x^j[k], y^j[k] \right) \quad \forall j \in \mathcal{N}_i^{\text{out}}(G_c)
\end{align*}
\]

(1)

where \( [k] \) denotes the \( k \)th iteration of the algorithm. The function \( f_i \) describes how agent \( i \) updates its estimate, while \( h_{ji} \) describes what information agent \( i \) communicates to agent \( j \).

In the PI setting, agent \( i \) cannot solve (NE-FI) because they do not know \( x_{-i} \). Therefore, consider the following problem for the PI setting:

\[
\begin{align*}
\min_{x^i} & \quad J_i \left( x^i, x_{-i}^i \right) \\
\text{s.t.} & \quad x^i \in \Omega_i \quad \text{(NE-PI-T)}
\end{align*}
\]

\[
\begin{align*}
x^i &= y^i \\
y^i &= x^i \quad \forall j \in \mathcal{N}_i^{\text{out}}(G_c)
\end{align*}
\]

(NE-PI-A)

Remark 2: Similar to the full-information setting case (see Remark 1), the NE can be characterized as the action profile \( x^* \) where the extended pseudogradient \( F(x) \triangleq \text{col}(\frac{\partial J_i(x)}{\partial x_i})_{i \in \mathcal{N}} \) is 0 and \( x \) is at consensus, e.g., \( F(x^*) = 0 \) and \( x^* = 1_N \otimes x^\star \) where \( x^\star \) is the NE.

Lemma 1: Given a game \( G \), let \( x^\star \) be a solution to (NE-FI) for all agents \( i \in \mathcal{N} \), i.e., \( x^\star \) is the NE. Assume that \( G_c \) is a strongly connected directed graph. If \( \bar{x} = (\bar{x}^1, \ldots, \bar{x}^N) \) solves (NE-PI-T) for all agents \( i \in \mathcal{N} \), then \( \bar{x} = 1_N \otimes x^\star \).

Proof: Since \( G_c \) is a strongly connected directed graph, the last two constraints in (NE-PI-T) are equivalent to the condition that \( x^i = x^j \) for all \( i, j \in \mathcal{N} \). Using the fact that \( x_{-i}^i = x_i \), this implies that \( x^i = x^j = x \) for all \( i, j \in \mathcal{N} \). Therefore, we can replace \( x^i \) with \( x_i \) and \( x^j \) with \( x_j \) in (NE-PI-T), which gives (NE-FI).

Remark 3: Lemma 1 shows that if each agent solves (NE-PI-T), then they will reach the NE. Notice that (NE-PI-T) introduces two additional constraints as compared with (NE-FI), which correspond to constraints on the communication between agents. The constraint \( x^i = y_{ij} \) means that \( i \)'s estimate is in agreement with the communicated message of all its neighbors. This can be interpreted as a local consensus constraint. The constraint \( y^i = x^i \) means that agent \( i \)'s communicated message to his out-neighbors is the same as its estimate, that is, agent \( i \) is communicating honestly with its neighbors.

While ideally we would like all the agents to solve (NE-PI-T), a self-interested agent has an incentive to ignore the second constraint (communicating honestly with its neighbors) if it would result in lowering their cost further. Thus, a self-interested agent would want instead to solve the following modified optimization problem:

\[
\begin{align*}
\min_{x^i} & \quad J_i \left( x^i, x_{-i}^i \right) \\
\text{s.t.} & \quad x^i \in \Omega_i \quad \text{(NE-PI-A)}
\end{align*}
\]

\[
\begin{align*}
x^i &= y^i \quad \forall j \in \mathcal{N}_i^{\text{out}}(G_c)
\end{align*}
\]

(NE-PI-A)

Note that (NE-PI-A) is obtained from (NE-PI-T) by removing the second constraint and modifying the first constraint. The constraint in (NE-PI-A) ensures that their belief about other agent’s actions matches what their neighbors communicate. If the agent is lying about their own action, then they do not want their action to match the lie, that is why \( x_{-i}^i = y^i_{-i} \) and not \( x^i = y^i_{-i} \).

Definition 3: Agent \( i \) is truthful if at each iteration \( k \), the message agent \( i \) is communicating is

\[
y^i[k] = x^i[k] \quad \forall j \in \mathcal{N}_i^{\text{out}}(G_c)
\]

(2)

and is adversarial otherwise.

Remark 4: When an agent is truthful, then the second equality constraint in (NE-PI-T) is immediately satisfied. Agents that intentionally send misinformation by not satisfying (2) are adversarial. In addition, we classify an agent that is trying to solve (NE-PI-T) but has their communication interfered with as adversarial because it appears to other agents that they are sending misinformation. This includes external factors to the game such as man-in-the-middle attacks, broken communication devices, signal jammers, etc [26]. For example, agent \( i \) is solving (NE-PI-T), but \( y^i[k] = 1 \forall k = \{0, 1, \ldots \} \), could represent that agent \( i \)'s communication device is broken.

Objective: Given a game \( G \), let \( x^\star \) denote the NE, and let \( \mathcal{N} = T \cup A \), where \( T \) is the set of truthful agents and \( A \) is the set of adversarial agents. Design an algorithm such that \( x^i \rightarrow x_i^\star \) for all \( i \in \mathcal{N} \), when the set \( A \neq \emptyset \).

In the literature, typical distributed NE seeking algorithms solve the problem where \( A = \emptyset \), and all agents satisfy (2). This is the standard formulation for distributed NE seeking problems, and by Lemma 1, the solution is the NE.

Remark 4: The optimization problem (NE-PI-T) or (NE-PI-A) is the problem agents intend to solve. In what follows, agents will filter out some of the communicated messages (of hopefully the adversarial agents) to learn the true action of agents and not be deceived.

III. OBSERVATION, ROBUST, AND INFORMATION ROBUST GRAPHS

In this section, we first discuss the issues of the current problem formulation and then introduce the observation graph to deal with this problem. The rest of this section will present various graph notions that are critical in the analysis of our proposed algorithm in the following section.

In the robust distributed optimization literature, it has been shown that, in general, it is impossible to detect an adversarial agent [19, Th. 4.4]. This is because an adversarial agent’s behavior is indistinguishable from a truthful agent with a modified (and valid) cost function.
For the NE seeking problem in the PI setting, there is additional information that could be used to detect adversarial agents. For example, assume that agent $i$ can measure/receive the actual/realized cost $J_i(x)$, which depends on the actual actions played, and agent $i$’s estimate $x^t_i$ has reached a steady-state value. Agent $i$ can compare the actual cost versus the estimated cost $J_i(x^t_i)$; if $J_i(x) \neq J_i(x^t_i)$, then agent $i$ knows that $x \neq x^t_i$ and the information they received has been tampered with. Therefore, agents would be able to detect that there is an adversarial agent, but they would not be able to identify which one of their opponents is adversarial. 

The core issue is that agents have no way of verifying if the information they receive from a particular agent is correct. In addition, if the information is determined to be incorrect, there is no mechanism to get the correct information. Unfortunately, without additional assumptions, it is impossible to solve this problem. In order to deal with these issues, we assume that each agent $i$ can directly observe/measure some of actions $x_j$, without communication over $G_c$.

**Definition 4:** An observation graph $G_o = (N, E_o)$ describes what actions can be directly observed.\(^1\) If edge $e_{ij} \in E_o$, then agent $i$ can directly observe/measure $x_j$. Note that agents now get information from two different sources. Agents get information via the communication graph $G_c$, which contains potentially incorrect information about all agents’ actions, and the observation graph $G_o$, which contains always correct local information about neighboring agents’ actions.

Continuing with the drone example, assume that the position of nearby drones can be measured (observation graph). The position of distant drones cannot be measured directly, but the position information for those drones is obtained through a communication graph, which can be susceptible to communication failures and signal interference.

Unless $G_o$ is a complete graph, agents will not fully know $x_{-i}$, i.e., agent $i$ does not know the actions of agents $\mathcal{N}_i \setminus \mathcal{N}_i^o(G_o)$. For agent $i$ to learn the actions of agents who are not observed, they will have to rely on the communication graph. Even if agents can directly observe a subset of the agents’ actions, the problem is not trivial. For example, consider that both $G_c$ and $G_o$ are given by the graph in Fig. 1.

The red (green) node in the graph represents the adversarial (truthful) agent. For this observation graph, agent 1 can measure $x_2$ and $x_3$, agent 2 can measure $x_4$, and agent 3 can measure $x_4$. Therefore, agent 1 does not know the action of agent 4, but agent 1’s neighbors know the action of agent 4. Using the communication graph $G_c$, agent 1 will receive two messages $y_{41}^2$ and $y_{41}^3$ about agent 4’s action: one truthful message and the other an adversarial message. It is not obvious how agent 1 can discern which of the two messages is truthful.

In the following, we analyze the properties of both the communication and observation graph. Under appropriate conditions on these two graphs, we will later prove the convergence of the proposed algorithm. The following definitions are used to describe how connected a subset of nodes to the rest of the graph is, and how connected a graph is.

**Definition 5 (see [11, Def. 4]):** Given a graph $G = (N, E)$, a subset $S \subseteq N$ is $r$-local if $(\forall i \in S, |N_i^o(G) \cap S| \leq r)$.\(^2\)

If the adversarial agent set $A$ is $r$-local, then this means that the number of adversarial neighbors to a truthful agent $i \in T$ is less than or equal to $r$. For example, in Fig. 2, the $a_i$ nodes are elements in $A$ and the numbered nodes are elements in $T$. The number represents how many in-neighbors are there in $A$. In this example, we can see that the most surrounded node has three neighbors in $A$; therefore, the set $A$ is 3-local.

**Definition 6:** Given a communication graph $G_c = (N, E_c)$ and an observation graph $G_o = (N, E_o)$, we say that node $i \in N$ is $r$-information robust\(^3\) if $(\forall S \subseteq N^o_i(G_o)) (\exists j \in S)$ such that $|N_j^o(G_c) \cap S| \geq r$.

A node $i$ being $r$-information robust describes how many paths are there from node $i$ to every other node. Fig. 3 shows one possible $S$ (nodes in the rectangle) that contains node $i$ and agents that can directly observe $i$, i.e., nodes $o_j$. You can interpret the set $S$ as all nodes $s \in S$ that have a path from $i$ to $s$. Node $i$ is $r$-information robust if there is a node $j$ outside of $S$ that has at least $r$ edges connecting into $S$. In Fig. 3, node $i$ is 2-information robust, since there is a node $j$ that has at least two edges connecting into $S$.\(^3\)

---

\(^1\)Unless there is only one other agent or the game has strong assumptions on the cost functions and the communication graph

\(^2\)In our setting, agent $i$ knows what action they used, and therefore, $e_{ii} \in E_o$ for all $i \in N$.

\(^3\)Node $i$ being $r$-information robust is related to the notion of an $r$-robust graph in [11].
The following lemma shows that if node \( i \) is \( r \)-information robust, then if at each node \( r - 1 \) in-edges are removed, the resulting graph is rooted at \( i \).

**Lemma 2:** Given a communication graph \( G_c = (\mathcal{N}, \mathcal{E}_c) \) and an observation graph \( G_o = (\mathcal{N}, \mathcal{E}_o) \), assume that node \( i \) is \( r \)-information robust. Let \( \mathcal{E}_o \) be the set of edges after \( (r - 1) \) in-edges from each node \( j \in \mathcal{N} \) have been removed from \( \mathcal{E}_c \). Then, the graph \( G = (\mathcal{N}, \mathcal{E}_o \cup \mathcal{E}_o) \) is rooted at \( i \).

**Proof:** Since \( \mathcal{E}_o \) is contained in the edge set of \( G \), there is an edge from node \( i \) to the nodes \( \mathcal{N}_{\text{out}}(G_o) \). Let \( X(i) \) denote all nodes in \( G \) that have a path from \( \mathcal{N}_{\text{out}}(G_o) \) to them. Therefore, \( Z(i) = \mathcal{N}_{\text{out}}(G_o) \cup X(i) \) is all nodes that have a path from \( i \).

We will show that \( Z(i) = \mathcal{N} \). Assume that the set \( Z(i) \) is not empty and is depicted in Fig. 4. Note that the set \( Z(i) \cap \mathcal{N} \text{out}(G_o) \) is not empty. By the assumption that node \( i \) is \( r \)-information robust, there must be a node \( j \in Z(i) \) that has at least \( r \) edges in \( \mathcal{E}_c \) from node \( i \). After removing \((r - 1)\) in-edges from node \( j \), there must be at least one edge going from \( Z(i) \) to \( Z(i) \). Therefore, node \( j \) should have been in \( Z(i) \) since there is a path from \( j \) to \( Z(i) \). Repeating this process will remove all the nodes from \( Z(i) \). Therefore, our assumption that \( Z(i) \) is not empty is false, \( Z(i) = \mathcal{N} \), and there is a path from \( i \) to any node on the graph, i.e., \( G \) is rooted at \( i \).

### IV. PROPOSED ALGORITHM

In this section, we present our proposed algorithm for NE seeking with truthful and adversarial agents. We make the following assumption about the number of adversarial agents that communicate with truthful agents.

**Assumption 2:** For the communication graph \( G_c \), the set \( \mathcal{A} \) is a \( D \)-local set and \((\forall i \in \mathcal{T}), |\mathcal{N}_{\text{t}}^i(G_c)| \geq 2D + 1 \).

This assumption states that around every truthful agent, there should be more truthful agents than adversarial ones, i.e., that a truthful agent \( i \) receives communicated messages from at most \( D \) adversarial agents and at least \( D + 1 \) truthful neighbors.

The algorithm is based on the idea of using some intermediary estimate denoted by \( v^i \). In constructing \( v^i \), agent \( i \) uses the actions that can be directly observed via \( G_o \), while relying on communicated messages sent for different actions that cannot be directly observed via \( G_c \). In constructing \( v^i \) for actions that are not directly observed, agent \( i \) removes/prunes extreme data values received from neighbors and takes a weighted average on the remaining ones. After the estimate \( v^i \) is constructed, it is used by agent \( i \) in a gradient step to update their own action toward minimizing their cost.

**Algorithm 1:** Robust NE Seeking Algorithm.

1. At any iteration \( k \), each agent \( i \in \mathcal{N} \) sends \( y^{ij}[k] \) to all \( j \in \mathcal{N}_{\text{out}}^i(G_c) \), and receives \( y^{ji}[k] \) from all \( j \in \mathcal{N}_{\text{in}}^i(G_c) \).

2. For each \( m \in \mathcal{N} \) and \( q \in \{1, 2, \ldots, n_m \} \), agent \( i \in \mathcal{T} \) removes the \( D \) highest and \( D \) smallest values \( y^{ij}[q] \) that are larger and smaller than its own value \( x^{ij} \), respectively (ties broken arbitrarily). If there are fewer than \( D \) values higher (respectively, lower) than its own value, \( i \) removes all of those values. \( y^{ij}[q] \) is the set of agents whose message about the \( q \)th component of \( m \)’s action is retained by \( i \) and agent \( i \) itself. Then, \( \forall q \in \{1, 2, \ldots, n_m \} \)

\[
\begin{align*}
\mathbf{v}^i_m[k] = \begin{cases} \\
\sum_{j \in \mathcal{N}_m} w_{mj}^i y^{ij}_m[k], & \text{if } m \notin \mathcal{N}^m_i(G_o) \\
x_{mq}^i[k], & \text{else}
\end{cases}
\end{align*}
\]

where the weights \( w_{mj}^i \) are such that for all \( i, j \in \mathcal{N} \), \( \sum_{j \in \mathcal{N}} w_{mj}^i[k] = 1 \); for all \( i \in \mathcal{N} \) and \( j \notin \mathcal{N}_{\text{out}}^i(G_o) \), \( w_{mj}^i[k] = 0 \); and there \( \exists \eta > 0, \forall i \in \mathcal{N} \) and \( j \in \mathcal{N}_{\text{out}}^i(G_o) \), \( w_{mj}^i[k] \geq \eta \).

3. Agent \( i \in \mathcal{T} \) updates its estimate vector as

\[
x^i[k + 1] = v^i[k] - \alpha \mathcal{R}_i^{\mathcal{T}} \frac{\partial J_i(v^i[k])}{\partial x_i}
\]

where \( \mathcal{R}_i = [0_{n_i \times n_i'} I_{n_i}, 0_{n_i \times n_i'}], \)
\( n_i' = \sum_{j < i} n_j, n_i'' = \sum_{j > i} n_j, \)
\( v_m^i = \text{col}(v_m^i)_{k \in \{1, 2, \ldots, n_m\}}, \) and
\( v^i = \text{col}(v^i_m)_{m \in \mathcal{N}}. \)

4. Agent \( i \in \mathcal{T} \) updates its communicated message

\[y^{ij}[k + 1] = x^i[k + 1] \]

In the above, \( v_{mq}^i \) is the intermediate estimate of agent \( i \) on the \( q \)th component of agent \( m \)’s action, while \( w_{mj}^i \in [0, 1] \) is the weight that agent \( i \) places on information from agent \( j \) about the \( q \)th component of agent \( m \)’s action. Step 2 is a filtering process, where for each message \( y_{mq}^i \) removed, the corresponding edge in \( G_c \) is used by agent \( i \) is removed (set \( w_{mj}^i = 0 \)). Notice that the filtering process results in an algorithm with a switching/time-varying communication graph.

**Remark 5:** The algorithm is inspired by [19] and [27], which use a pruning plus averaging step followed by a gradient step for a distributed optimization problem. However, there are differences in the setting, analysis, and results. The first major difference is that the authors in [19] and [27] only focused on finding the minimizer of the set of truthful agents in the network and ignored the actions of the adversarial agents. In addition, Sundaram and Gharesifard [19] ignored the dynamics of the adversarial agents. For the NE problem, the agents’ cost functions are dependent on the actions of the adversarial agents and, therefore, cannot be ignored. We seek to find the NE, \( x^* \), which includes both truthful and adversarial agent’s actions. Furthermore, every \( j \in \mathcal{A} \) has an optimization problem it wants to solve, which is coupled to the actions/estimates of
the others. Therefore, we also need to model how adversarial agents update their estimates/actions. Second, in [19] and [27], a solution in the convex hull of the local minimizers is found, but this is not guaranteed to be a global minimizer. Herein, we converge to the NE and not just within a neighborhood of the NE. Third, the authors in [19] and [27] assume that each agent’s function is convex with respect to $x$ and Lipschitz. Herein, we assume that $F$ is strongly monotone and Lipschitz. The assumption on $F$ is significantly different from the ones used in distributed optimization, making the analysis in [19] and [27] not applicable for NE problems. Finally, we introduce an observation graph that is not present in [19] and [27]. This makes the NE problem tractable and allows us to prove convergence to the NE.

Since adversarial agents are motivated to solve (NE-Pl-A) and use their messages to deceive other agents and to avoid being deceived, we make the following assumption that characterizes this behavior.

**Assumption 3:** For each $i \in \mathcal{A}$, agent $i$’s estimate $v^i$ can be expressed as a weighted average of only truthful agents’ messages, and agent $i$ updates $x^i$ via step 3 of Algorithm 1.

**Remark 6:** We justify Assumption 3 by first noting that each agent $i \in \mathcal{A}$ solves an optimization problem, and gradient descent-type dynamics is a common method used. The assumption that agent $i$ updates “$v^i$ as a function of just truthful agents’ messages” obviously holds when $|\mathcal{A}| = 1$. When $|\mathcal{A}| > 1$, we argue that each agent $i \in \mathcal{A}$ tries to solve its own optimization problem and, therefore, needs to know the actions of the other agents. It only seems reasonable that if agent $i$ tries to deceive the other agents, that agent $i$ would also protect itself from being deceived by others. Here are some instances where this assumption holds:

1) if agents create a coalition $C$ and spread misinformation to agents in $C$. Then, all the agents in $C$ would know which information can be trusted;
2) if the adversarial agents are spread out and do not directly communicate to each other;
3) if adversarial agents are able to get access to all the actions $x_j$ directly;
4) if agent $i$’s cost function is $J_i(x) = 0$ and only wants to disrupt the system;
5) if agent $i$ uses step 2 of Algorithm 1 to filter out other adversarial agents as a method to protect itself. The following lemma shows that step 2 can be equivalently expressed as a weighted average of only truthful agents’ messages.

### V. CONVERGENCE ANALYSIS

To prove that Algorithm 1 converges to the NE, we first show that step 2 of the algorithm can be equivalently written in terms of truthful agents only. Note that this representation is only used for analysis; agents do not actually communicate to only truthful agents.

**Lemma 3:** Consider the observation graph $G_o$ and communication $G_c$ under Assumption 2. Then $\forall m \in \mathcal{N}$, $\forall q \in \{1, 2, \ldots, n_m\}$ $\forall i \in \mathcal{T}$, (3) can be written as

$$v^i_{mq}[k] = \begin{cases} \sum_{j \in \mathcal{N}_m^t} w^{ij}_{mq} [k] y^j_{mq} [k] , & \text{if } m \notin N^t_i(G_o) \\ \{x_{mq} [k], & \text{else} \end{cases}$$

(4)

where $w^{ij}_{mq}$ are new weights and $Y^t_{mq} \subset (N^t_i(G_o) \cap \mathcal{T}) \cup \{i\}$ are truthful agents whose message is retained. The weights $w^{ij}_{mq}$ are such that for all $i, j \in \mathcal{N}$, $\sum_{j \in \mathcal{N}} w^{ij}_{mq} [k] = 1$; for all $i \in \mathcal{N}$ and $j \notin \mathcal{N}^t_i$, $w^{ij}_{mq} [k] = 0$; $\forall i \in \mathcal{N}$ and $j \in \mathcal{N}_m^t$, $w^{ij}_{mq} [k] > 0$; and $\forall i \in \mathcal{N}$, there is at least $|N^t_i(G_o)| - 2D$ nodes $j \in \mathcal{Y}^t_{mq}$ with $w^{ij}_{mq} \geq \frac{\alpha}{2}$.

**Proof:** Note that the difference between (3) and (4) is that the set $\mathcal{Y}^t_{mq}$ and the weights $w^{ij}_{mq}$ are replaced with $\mathcal{Y}^t_{mq}$ and $\tilde{w}^{ij}_{mq}$ respectively. The set $\mathcal{Y}^t_{mq}$ is the set of agents whose message is retained, which can contain adversarial agents, while the set $\mathcal{Y}^t_{mq}$ is the set of only truthful agents retained.

Since $v^i_{mq}$ is the same for both (3) and (4) when $m \in N^t_i(G_o)$, we only need to consider $m \notin N^t_i(G_o)$, i.e., agents unobserved by $i$. The rest of the proof follows the same logic as [19, Proposition 5.1]. The idea of the proof is that either all adversaries are filtered out ($\mathcal{Y}^t_{mq} = \mathcal{Y}^t_{mq}$) or messages from adversaries that are not filtered out can be written as a linear combination of agents in $\mathcal{T}$. This is because if $y^{ij}_{mq}$ for $j \notin \mathcal{A}$ is not filtered out, then there exists $s, t \in \mathcal{T}$ such that $y^{is}_{mq} \leq y^{ij}_{mq} \leq y^{it}_{mq}$ and $y^{ij}_{mq} = (1-\alpha)y^{is}_{mq} + \alpha y^{it}_{mq}$ for some $\alpha \in [0, 1]$. The weight $w^{ij}_{mq}$ of the adversarial node would be added to the weights of $w^{is}_{mq}$ and $w^{it}_{mq}$ by (1-\alpha)\tilde{w}^{ij}_{mq}$ and $\alpha \tilde{w}^{ij}_{mq}$ respectively, resulting in $\tilde{w}^{is}_{mq}$ and $\tilde{w}^{it}_{mq}$.

Notice that under Assumption 3, adversarial agent $i \in \mathcal{A}$ can update $v^i_{mq}$ as a function of just truthful messages. Similarly, by Lemma 3, truthful agents $i \in \mathcal{T}$ can also update $v^i_{mq}$ as a function of just truthful messages. In addition, since truthful messages must satisfy (2), we can express $v^i_{mq}$ as a function of $x^i_{mq}$. For agent $i \in \mathcal{T}$, let $x^i_{mq} = \text{row}(\tilde{W}^{ij}_{mq})_{j \in \mathcal{N}}$, where $\tilde{w}^{ij}_{mq}$ are the weights from Lemma 3, and let $x_{mq} = \text{col}(x^i_{mq})_{j \in \mathcal{N}}$. Then, we can compactly write $v^i_{mq} = \tilde{w}^i_{mq} x_{mq}$ when $m \notin N^t_i(G_o)$. With abuse of notation for adversarial agent $i \in \mathcal{A}$, let $\tilde{w}^i_{mq}$ denote the weights that adversarial agents place on the messages received. Then, adversarial agents can also write $v^i_{mq} = \tilde{w}^i_{mq} x_{mq}$ when $m \notin N^t_i(G_o)$. Let $E_{mq} \in \mathbb{R}^{n_m \times n}$ be a diagonal matrix with the $(n_m^c + q)$-diagonal entry equal to 1 and 0 elsewhere, $\tilde{W}^i = \sum_{m \in \mathcal{N}} \sum_{q \in \{1, 2, \ldots, n_m\}} \tilde{w}^{ij}_{mq} \otimes E_{mq}$, and $O^t \in \mathbb{R}^{n \times n}$ be a diagonal matrix where the $(n_m^c + q)$-diagonal entry is 1 if agent $i$ can observe $m$ and 0 otherwise. Then, for agent $i$, we can compactly write

$$v^i = (I - O^t) \tilde{W}^i x + O^t x.$$  

(5)

The following lemma gives a compact form for Algorithm 1:

**Lemma 4:** Under Assumptions 2 and 3, Algorithm 1 is equivalent to

$$v[k] = \tilde{W}[k] x[k]$$  

(6)

$$x[k + 1] = v[k] - \alpha R^T F(v[k])$$  

(7)
Fig. 5. Graph of $G_o = G_o$.

where $\mathcal{R} = \text{diag}(\mathcal{R}_i)_{i \in \mathcal{N}}$, $\mathbf{F}(\mathbf{x}) = \text{col}(\frac{\partial f_i(\mathbf{x})}{\partial x_i})_{i \in \mathcal{N}}$ is the extended pseudogradient, $\mathbf{v} = \text{col}(\mathbf{v}_i)_{i \in \mathcal{N}}$, and

$$\mathbf{W}[k] = (I - O) \tilde{\mathbf{W}}[k] + O \quad (8)$$

where $O = \text{col}(O^i)_{i \in \mathcal{N}}$ and $\tilde{\mathbf{W}}[k] = \text{col}(\tilde{W}_i[k])_{i \in \mathcal{N}}$.

**Proof:** It is trivially obtained by stacking (5) and step 3 of Algorithm 1. Note that step 3 also applies to adversarial agents by Assumption 3.

**Remark 7:** Note that (7) in Lemma 4 represents an NE seeking algorithm that uses a communication step followed by a gradient step (with constant step size). However, the matrix $\mathbf{W}[k]$ is a time-varying row stochastic rooted matrix, which is not from a finite or countable collection of stochastic matrices. We are unaware of any other NE seeking algorithm that can deal with this case.

While Lemma 3 does show that step 2 of the algorithm can be described using messages sent only by truthful agents, this is not enough to ensure that agents learn $x$. This is because we have not assumed anything on the structure of $\mathbf{W}[k]$. Fig. 5 shows a counterexample. Assume that $\mathcal{A} = \{j\}$; therefore, $\mathcal{A}$ is 1-local and $|\mathcal{N}_i^o(m)(G_c)| \geq 3$ and satisfies Assumption 2. However, it is obvious that agents will not learn $x$.

Node $j$ can always manipulate the information between node $p_1, p_2, \ldots$ and $o_1, o_2, \ldots$. We can see that the graph is not connected enough for information to flow throughout the network. To deal with this case, we make the following assumption.

**Assumption 4:** Given the communication graph $G_c$ and observation graph $G_o$, all nodes $m \in \mathcal{N}$ are $(2D + 1)$-information robust.

**Remark 8:** This assumption ensures that after the pruning process of step 2 in Algorithm 1, the information about any agent $m$ can reach all the agents in $\mathcal{N}$. In addition, note that the assumption cannot be relaxed. To understand why, first note that agents do not know who is truthful and who is adversarial and have to act the same way for both the sets of agents. Second, agents do not know if they can learn $x_i$ through direct communication with agent $i$, since agent $i$ could be adversarial. By the filtering process, up to $2D$ messages can be removed. Therefore, to ensure that the information from any set $S \supset \mathcal{N}_i^o(m)(G_o)$ propagates, there must be an agent $j$ that has $2D + 1$ edges connected to $S$; otherwise, all the messages from $S$ could potentially be filtered out, and then, the information will never spread through the network.

**Lemma 5:** Consider the communication graph $G_c$ and observation graph $G_o$, under Assumptions 2–4. Then, for each $m \in \mathcal{N}$ and $q \in \{1, 2, \ldots, n_m\}$, the graph representing the information exchange of the $q$th component of agent $m$’s action induced by (8) has a path from $m$ to every other node, i.e., $m$ is rooted. □

**Proof:** For each $m \in \mathcal{N}$ and $q \in \{1, 2, \ldots, n_m\}$, we know from Lemma 3 that each agent in $T$ has $|\mathcal{N}_m^o(G_c)| \geq 2D$. Therefore, each agent has removed at most $2D$ nodes. Since the graph is $(2D + 1)$-information robust, from Lemma 2, we know that there is at least one path from $m$ to every node in $T$. Since there is a path to every truthful agent and adversarial agents get their messages from only truthful agents, by Assumption 3, we know that there is a path to all the adversarial agents as well. □

**Remark 9:** In the literature for resilient consensus algorithms [11, Th. 2] and resilient distributed optimization assumption [19, Th. 6.1], Assumption 2 is a standard assumption. If Assumption 2 does not hold, then adversarial agents cannot be filtered out, and consensus cannot occur. In addition, the assumption that the graph is $(2D + 1)$-robust [11], which is very closely related to $(2D + 1)$-information robust (see Assumption 4), is a standard assumption that ensures that the graph is rooted after removing edges [11, Th. 2], [19, Th. 6.1].

Before proving that our proposed algorithm converges to the NE, we first highlight some important properties $\mathbf{W}[k]$ has under Assumptions 3 and 4. The matrix $\mathbf{W}[k]$ is a time-varying row stochastic matrix with $\mathbf{W}[k](1_N \otimes v) = 1_N \otimes v \forall k \geq 0$ and $\forall v \in \mathbb{R}^n$. In addition, $\mathbf{W}[k] = \mathbf{W}[k]$. If $A \tilde{=}_{N} \mathcal{R} \forall k \geq 0$, then $\mathbf{W}[k] = \mathbf{W}[k]$ is $A \tilde{=}_{N}$-robust (see Assumption 4), and $\mathbf{W}[k]$. The following Lemma gives an important property of the matrix $\mathbf{W}[k]$.

**Lemma 6:** Consider the communication graph $G_c$ and the observation graph $G_o$, under Assumptions 2–4. If $\mathbf{W}[k]$ is as in Lemma 4, then there exists a sequence of positive-semidefinite matrices $P[k]$ such that $\mathbf{W}[k]P[k] = P[k] - (I - A)^T(I - A)$

$$\mathbf{W}[k]P[k] = P[k] - (I - A)^T(I - A)$$

where $I$ is the identity matrix of appropriate dimensions, and $\text{Null}(P[k]) = 1 \otimes v$, for all $k \geq 0$, $v \in \mathbb{R}^n$. In addition, $P[k] \preceq \tilde{P}$, where $\tilde{P} = \frac{1}{1 - \frac{\|A\|}{\|\mathcal{R}\|}}$, with $C = 1 - (\frac{\|A\|}{\|\mathcal{R}\|})^{N-1}$. □

**Proof:** See the Appendix.

**Remark 10:** The upper bound $\tilde{P}$ is very conservative because it is obtained assuming the worst possible communication graph switching. In practice, $\tilde{P}$ is significantly lower.

**Theorem 1:** Given a game $G$, with communication graph $G_c$ and observation graph $G_o$ such that Assumptions 1–4 hold, let

$$M \tilde{=} \begin{bmatrix} 2\alpha\mu - \tilde{p}\alpha^2 L^2 & -\alpha\tilde{p}L (1 + \alpha L) \\ -\alpha\tilde{p}L (1 + \alpha L) & 1 - 2\alpha(1 - \tilde{p})L - \tilde{p}^2 L^2 \end{bmatrix} \quad (9)$$

with $\tilde{p}$ as in Lemma 6, and select $\alpha$ such that $M > 0$. Then, for any initial condition $x_0, v_0$, the iterates generated by Algorithm 1 are $\text{col}(x^k[i])(i \in \mathcal{N}) \rightarrow 1 \otimes x^*$, where $x^*$ is the NE.

**Proof:** By Lemma 4, the updates of Algorithm 1 are compactly written as (6) and (7). To simplify notation, in the following, we denote iteration $[k]$ with the subscript $k$ and $H \tilde{=} \mathcal{R}^T \mathbf{F}$. Then, from (6) and (7), $v_{k+1} = \mathbf{W}[k] (v_k - \alpha H v_k)$. Consider
the following candidate Lyapunov function:

$$V_{k+1} = \|v_{k+1} - Av_{k+1}\|^2_{P_{k+1}} + \|Av_{k+1} - x^*\|^2$$  \hspace{1cm} (10)

where $P_{k+1}$ is as in Lemma 6 and $x^* = 1 \otimes x^*$. It can be shown that $V_{k+1}$ is positive definite at $x^*$ and radially unbounded. Indeed, the first term in $V_{k+1}$ is zero only when $v_{k+1} = Av_{k+1}$, because the null space of $P_{k+1}$ is the null space of $(I - A)$ for all $k$ (see Lemma 6), i.e., when $v_{k+1}$ is equal to its projection onto the consensus subspace. Thus, $V_{k+1}$ is zero only when $\|Av_{k+1} - x^*\|^2 = \|v_{k+1} - x^*\|^2$, which implies $v_{k+1} = x^*$.

Note that $\nabla W_k = \nabla_k A = A$ for all $k$ and $\|I - A\| = 1$. In addition, from Lemma 6, we know that $P_k \preceq \bar{p}I$. Then, from (6) and (7), the first term in (10) to obtain

$$\|v_{k+1} - Av_{k+1}\|^2_{P_{k+1}} = \|(I - A)\nabla W_k x_{k+1}\|^2_{P_{k+1}} = \|(I - A)\nabla_k (I - A)x_{k+1}\|^2_{P_{k+1}} = \|(I - A)v_k - (I - A)(\alpha H)v_k\|^2_{P_{k-1}} = \|(I - A)v_k - (I - A)(Hv_k - HA v_k - Hx^*)\|^2_{P_{k-1}} = \|\nabla_k - \alpha(I - A)(Hv_k - HAv_k - HA v_k - Hx^*)\|^2_{P_{k-1}}$$

where we use $\nabla W_k = \nabla_k A$ and Lemma 6 on line 3. In addition, we use Remark 2 ($F(x^*) = 0$) to add $Hx^* = R^T F(x^*) = 0$. Expanding out the right-hand side, we obtain

$$\|v_{k+1} - Av_{k+1}\|^2_{P_{k+1}} = \|(I - A)v_k\|^2_{P_{k-1}} + \|\alpha(I - A)(Hv_k - HAv_k)\|^2_{P_{k-1}} + \alpha^2\|\nabla_k - \alpha(I - A)(Hv_k - HAv_k)\|^2_{P_{k-1}} - 2\alpha\langle(I - A)v_k, (I - A)(Hv_k - HAv_k)\rangle_{P_{k-1}}$$

Next, we use the fact that $\|a\|^2_{P_{k-1}} \leq (\bar{p} - 1)\|a\|^2$ and $\|\alpha - b\|^2_{P_{k-1}} \leq (\bar{p} - 1)\|\alpha - b\|^2$. Thus

$$\|v_{k+1} - Av_{k+1}\|^2_{P_{k+1}} \leq \|v_k - Av_k\|^2_{P_{k-1}} + \|\alpha - Av_k\|^2_{P_{k-1}} + \alpha^2\|\nabla_k - \alpha(I - A)(Hv_k - HAv_k)\|^2_{P_{k-1}} + 2\alpha\langle(I - A)v_k, (I - A)(Hv_k - HAv_k)\rangle_{P_{k-1}} - 2\alpha\|\alpha - Av_k\|^2_{P_{k-1}}$$

where we use the fact that $H$ is $L$-Lipschitz from Assumption 1. From (6) and (7), the second term in (10) is equal to

$$\|Av_{k+1} - x^*\|^2 = \|A(1 - \alpha H)v_k - (1 - \alpha H)x^*\|^2$$

and

$$\|Av_k - x^*\|^2 = \|A(Hv_k - HA v_k - HAv_k - Hx^*)\|^2$$

where $\|Av_{k+1} - x^*\|^2 = \|A(1 - \alpha H)v_k - (1 - \alpha H)x^*\|^2 + \alpha^2\|AHv_k - AHx^*\|^2 - 2\alpha\langle Av_k - x^*, Av_k - AHx^*\rangle - 2\alpha\langle Av_k - x^*, AHv_k - AHx^*\rangle + 2\alpha^2\|AHv_k - AHAv_k - AHx^*\|^2$.

Note that $A\nabla^T = I_N \otimes I_n$ and $Av_k = I_N \otimes v_k$, where

$$v_k = \text{col}\{v_{i,k}\}_{i \in N}, H = R^T F, \text{ and } F(1_N \otimes z) = F(z) \forall z \in R^n.$$ Then

$$\langle Av_k - x^*, AHv_k - AHx^*\rangle = \langle I_N \otimes v_k - I_N \otimes x^*, I_N \otimes F(I_N \otimes v_k) - I_N \otimes F(1_N \otimes x^*)\rangle$$

and

$$\langle I_N \otimes v_k - I_N \otimes x^*, I_N \otimes F(v_k) - I_N \otimes F(x^*)\rangle = N\|v_k - x^*\|^2 - \|F(v_k) - F(x^*)\|^2 \geq N\mu\|v_k - x^*\|^2$$

and bounding the remaining by using the fact that $\|A\| = 1$ and $H$ is $L$-Lipschitz, yields

$$\|Av_{k+1} - x^*\|^2 = \|Av_k - x^*\|^2 + \alpha^2\|Av_k - x^*\|^2 + \alpha^2\|Av_k - x^*\|^2$$

Combining the inequalities (11) and (14), $V_{k+1}$ is bounded by

$$\|v_{k+1} - v_k\| \leq \|v_k - \omega_k\|_M \omega_k^T, \text{ where } \omega_k = \text{col}\{\|Av_k - x^*\|, \|v_k - Av_k\|\} \text{ and } M \text{ is as in (9).}$$

Note that when $\omega_k = 0$, then $v_k$ is at consensus and $v_k = x^* = 1 \otimes x^*$. Hence, by [28, Th. 13.11], $v_k$ converges to $1 \otimes x^*$, and by continuity, from (7), $x_{k+1}$ converges to $1 \otimes x^*$, where $x^*$ is the NE.

Remark 11: Note that $M_{ij} > 0$ and $M_{22} > 0$ for small $\alpha$. In addition, $M_{11}M_{22} \propto \alpha$ and $M_{12}M_{21} \propto \alpha^2$. Therefore, there exists a sufficiently small $\alpha$ such that $M_{11}M_{22} > M_{12}M_{21}$ holds; hence, $M$ is positive definite.

While it is possible to obtain a linear convergence rate for Theorem 1, it has been omitted because the rate is dependent on $\bar{p}$, which is very conservative, and therefore, the rate will be significantly slower than that obtained experimentally.

VI. SIMULATION

In this section, we consider a sensor network/robot formation problem modeled as a game [29]. The objective of each agent is to be near a subset of agents while at the same time staying close to their prescribed location. We consider a group of mobile robots in the plane, where the cost function for each agent is $J_i(x_i, x|) = \alpha(x) + r_i(x)$, where $\alpha(x) = 1/2\|x/N \sum_{j \in N} x_j - Q\|^2$ is the distance of the average from the target $Q(= 0)$ location, and $r_i(x) = \sum_{j \in N^\circ} (g_{ij} \otimes 1/2\|x_i - x_j - d_{ij}\|^2$ quantifies the relative position cost with its neighbor $x_j$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
The graph $G_{\text{cost}}$ is induced by the solid edges in Fig. 6, and $d_{4,5} = (-1, 0)$ is the quadratic cost for agent 4 not being 1 unit to the left of 5. We assume that the adversarial agent $i$ is sending $y^{ji} = x^{ji} + \sigma(0, 1)$, where $\sigma(0, 1)$ is a vector of Gaussian noise with mean 0 and standard deviation 1.

We consider two examples that have different $G_c$ and $G_o$. In the first example, we assume that $G_c = G_o$ and there is an edge from $j$ to $i$ if $\|x_i - x_j\|_\infty \leq 2$ in Fig. 6. For example, the large blue circle in Fig. 6 communicates with all the agents inside the green square. In addition, the red nodes in Fig. 6 are the adversarial agents, the adversary set is 3-local, and $D = 3$. In the second example, $E_o \subset E_c$, where the solid lines in Fig. 6 are the edges of $E_o$, and the solid and dashed lines are edges of $E_c$. In addition, the red nodes in Fig. 6 are the adversarial agents, the adversary set is 1-local, and $D = 1$.

We note that the step sizes from Theorem 1 are very conservative (cf. Remark 10). We run Algorithm 1 with the step size $\alpha = 1/40$. Fig. 7 shows the distance to the NE, and Fig. 8 shows the position of a subset of the agents for Example 1 (Example 2 is omitted due to space constraints).

VII. CONCLUSION

In this article, we considered the NE seeking problem in the PI setting where agents are adversarial. We designed an algorithm that is robust against adversarial agents by utilizing an observation graph. The agents prune out the largest/smallest elements, take a weighted average of the remaining elements, and then perform a gradient step. Under appropriate assumptions about the number of adversarial agents and the connectivity of the graph, we are able to prove convergence to the NE. A possible extension to this work would be to design a resilient generalized NE seeking algorithm or relax some of the assumptions (e.g., Assumption 3).

APPENDIX

Proof of Lemma 6

Let $\Phi(k + s, k) \triangleq \overline{W}[k + s]\overline{W}[k + s - 1] \cdots \overline{W}[k]$ for all $s \geq 0$ and $\Phi(k - 1, k) \triangleq I$. Let $\Gamma(k + s, k) \triangleq \Phi(k + s, k)^T \Phi(k + s, k)$. Let $P[k] \triangleq (I - A)^T(\sum_{j=k}^{\infty} \Gamma(j - 1, k))(I - A)$.

We can see that

\[
\overline{W}[k]^T P[k + 1] \overline{W}[k] = \overline{W}[k]^T (I - A)^T \left( \sum_{j=k+1}^{\infty} \Gamma(j - 1, k + 1) \right) (I - A) \overline{W}[k] \\
= (I - A)^T \overline{W}[k]^T \left( \sum_{j=k+1}^{\infty} \Gamma(j - 1, k + 1) \right) \overline{W}[k] (I - A) \\
= (I - A)^T \left( \sum_{j=k}^{\infty} \Gamma(j - 1, k) - I \right) (I - A) \\
= P[k] - (I - A)^T(I - A).
\]

Note that $\Gamma(j - 1, k)$ has full rank because the graph $\overline{W}[k]$ is strongly connected for all $k$ and $\text{Null}(I - A) = 1 \otimes v$ for all $v$. Therefore, $\text{Null}(P[k]) = 1 \otimes v$.

Next, we find an upper bound on $P[k]$, i.e., $P[k] \preceq \bar{P} I$. The following argument holds for all $m \in N$ and $q \in \{1, 2, \ldots, n_m\}$. Notice that the elements in $\Phi(k + s, k)$ are the sum of all weighted paths from node $i$ to node $j$ about action $m$. From Lemma 5, we know that for node $m$, the communication graph is induced from $\tilde{w}_{mq}^{10}$ and the observation graph $G_o$ is $1$-information robust for all $k$ and the edges have weight of at least $\eta/2$. Therefore, there exists a path, with at most $N - 1$ edges, from $m$ to every node. The element $\phi_{mq}^{im}$ in $\Phi(k + N - 1, k)$, representing the path from $m$ to any node $i$ about component $q$ of action $m$, has a weight of at least $1$.
\((\eta/2)^{N-1} = 1 - C\) for any iteration \(k\).

Let \(\hat{\phi}_{mq}^{im}\) be an element in \(\Phi(k + 2(N - 1), k + 1)\) and \(\tilde{\phi}_{mq}^{im}\) in \(\Phi(k + 2(N - 1), k + 1)\). Then

\[
\hat{\phi}_{mq}^{im} = \sum_{j=0}^{N} \delta_{ij}^{mq} \phi_{mq}^{im} = \hat{\phi}_{mq}^{im} \phi_{mq}^{mm} + \sum_{j=0, j\neq m}^{N} \delta_{ij}^{mq} \phi_{mq}^{m}\]

\[
\geq \left( \frac{\eta}{2} \right)^{N-1} \left( 1 - \frac{\eta}{2} \right)^{N-1} \left( \frac{\eta}{2} \right)^{N-1} = 1 - C^2\]

where we used the fact that \(\phi_{mq}^{mm} = 1\), \(\hat{\phi}_{mq}^{im} \phi_{mq}^{m} \geq \left( \frac{\eta}{2} \right)^{N-1}\), and \(\sum_{j=0}^{N} \delta_{ij}^{mq} = 1\). By an inductive argument, it can be shown that the weight \(\hat{\phi}_{mq}^{im}\) in \(\Phi(k + r(N - 1), k)\) is at least \((1 - C^r)\) for all \(m \in N\). Note that \(\Phi(k + r(N - 1), k) - A\) by the properties of \(A\) and \(W[k]\), and the rows are of the form

\[
\begin{bmatrix}
\hat{\phi}_{mq}^{im}, \ldots, \hat{\phi}_{mq}^{im-1}, \hat{\phi}_{mq}^{im} - 1, \hat{\phi}_{mq}^{im+1}, \ldots, \hat{\phi}_{mq}^{imN}.
\end{bmatrix}
\]

Therefore, we know that the infinity norm is

\[
\|\Phi(k + r(N - 1), k) - A\|_{\infty} = 2C^r
\]

Using this bound, we know that

\[
\|P[k]\| = \left\| \sum_{j=k}^{\infty} (I - A)^T \Gamma(j - 1, k)(I - A) \right\|
\]

\[
\leq \sum_{j=k}^{\infty} \| (I - A)^T \Gamma(j - 1, k)(I - A) \|
\]

\[
\leq N \sum_{j=k}^{\infty} \| \Phi(j - 1, k)(I - A) \|_2^2
\]

\[
= 4N \sum_{j=k}^{\infty} C^{2(j-k+1)-1} = \frac{4N}{1 - C^{2N}}
\]

where the second last line follows from \(\| M \|_2 \leq \sqrt{\eta} \| M \|_{\infty}\), \(M \in \mathbb{R}^{m\times n}\), and the fact that \(\Phi\) is made of \(n\) independent graphs of size \(N\).

REFERENCES

[1] W. Saad, Z. Han, H. Poor, and T. Başar, “Game theoretic methods for the smart grid,” IEEE Signal Process. Mag., vol. 29, no. 5, pp. 86–105, Sep. 2012.

[2] F. Parise, M. Colombino, S. Grammatico, and J. Lygeros, “Mean field constrained charging policy for large populations of plug-in electric vehicles,” in Proc. IEEE Conf. Decis. Control, 2014, pp. 5101–5106.

[3] N. Li, L. Chen, and M. Dahleh, “Demand response using linear supply function bidding,” IEEE Trans. Smart Grid, vol. 6, no. 4, pp. 1827–1838, Jul. 2015.

[4] J. Barrera and A. Garcia, “Dynamic incentives for congestion control,” IEEE Trans. Autom. Control, vol. 60, no. 2, pp. 299–310, Feb. 2015.

[5] G. Scutari, F. Facchinei, J.-S. Pang, and D. P. Palomar, “Real and complex monotone communication games,” IEEE Trans. Inf. Theory, vol. 60, no. 7, pp. 4197–4231, Jul. 2014.

[6] F. Facchinei and J. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems. New York, NY, USA: Springer, 2007.

[7] G. Belgioioso and S. Grammatico, “A Douglas-Rachford splitting for semi-decentralized equilibrium seeking in generalized aggregative games,” in Proc. IEEE 57th Conf. Decis. Control, 2018, pp. 3541–3546.

[8] D. Gadjov and L. Pavel, “A passivity-based approach to Nash equilibrium seeking over networks,” IEEE Trans. Autom. Control, vol. 64, no. 3, pp. 1077–1092, Mar. 2019.

[9] T. Tatarenko, W. Shi, and A. Nedich, “Geometric convergence of gradient play algorithms for distributed Nash equilibrium seeking,” IEEE Trans. Autom. Control, vol. 66, no. 11, pp. 5342–5353, Nov. 2021.

[10] M. Bianchi and S. Grammatico, “Fully distributed Nash equilibrium seeking over time-varying communication networks with linear convergence rate,” IEEE Control Syst. Lett., vol. 5, no. 2, pp. 499–504, Apr. 2021.

[11] H. J. Leblanc, H. Zhang, X. Koutsoukos, and S. Sundaram, “Resilient asymptotic consensus in robust networks,” IEEE J. Sel. Areas Commun., vol. 31, no. 4, pp. 766–781, Apr. 2013.

[12] Y. Wang, H. Ishii, F. Bonnet, and X. Dfego, “Resilient real-valued consensus in spite of mobile malicious agents on directed graphs,” IEEE Trans. Parallel Distrib. Syst., vol. 33, no. 3, pp. 586–603, Mar. 2022.

[13] J. Usevitch and D. Panagou, “Resilient leader-follower consensus with time-varying leaders in discrete-time systems,” in Proc. IEEE 58th Conf. Decis. Control, 2019, pp. 5432–5437.

[14] D. Senejohnny, S. Sundaram, C. De Persis, and P. Tesi, “Resilience against misbehaving nodes in self-triggered coordination networks,” in Proc. IEEE Conf. Decis. Control, 2018, pp. 2848–2853.

[15] Z. Li, Z. Duan, and F. L. Lewis, “Distributed robust consensus control of multi-agent systems with heterogeneous matching uncertainties,” Automatica, vol. 50, no. 3, pp. 883–889, 2014.

[16] C. Zhao, J. He, and J. Chen, “Resilient consensus with mobile detectors against malicious attacks,” IEEE Trans. Signal Inf. Process. Netw., vol. 4, no. 1, pp. 60–69, Mar. 2018.

[17] Y. Chen, S. Kar, and J. M. Moura, “Attack resilient distributed estimation: A consensus innovations approach,” in Proc. Ann. Amer. Control Conf., 2018, pp. 1015–1020.

[18] L. Su and N. H. Vaidya, “Fault-tolerant multi-agent optimization: Optimal iterative distributed algorithms,” in Proc. ACM Symp. Princ. Distrib. Comput., 2016, pp. 425–434.

[19] S. Sundaram and B. Gharesifard, “Distributed optimization under adversarial nodes,” IEEE Trans. Autom. Control, vol. 64, no. 3, pp. 1063–1076, Mar. 2019.

[20] W. Fu, Q. Ma, J. Qin, and Y. Kang, “Resilient consensus-based distributed optimization under deception attacks,” Int. J. Robust Nonlinear Control, vol. 31, no. 6, pp. 1803–1816, 2021.

[21] K. Kuwaranancharoen, L. Xin, and S. Sundaram, “Byzantine-resilient distributed optimization of multi-dimensional functions,” in Proc. Amer. Control Conf., 2020, pp. 4399–4404.

[22] T. Moscibroda, S. Schmid, and R. Wattenhofer, “The price of malice: A game-theoretic framework for malicious behavior,” Internet Math., vol. 6, no. 2, pp. 125–156, 2009.

[23] T. Zhang and Q. Zhu, “A game-theoretic foundation of deception: Knowledge acquisition and fundamental limits,” 2018, arXiv:1810.00072.

[24] Z. Feng and G. Hu, “Attack-resilient distributed algorithms for exponential Nash equilibrium seeking,” 2020, arXiv:2009.10666.

[25] D. Gadjov and L. Pavel, “Distributed Nash equilibrium seeking resilient to adversaries,” in Proc. IEEE 66th Conf. Decis. Control, 2021, pp. 191–196.

[26] J. Pawlick and Q. Zhu, “Game Theory for Cyber Deception,” Cham, Switzerland: Birkhäuser, 2021.

[27] C. Zhao, J. He, and Q.-G. Wang, “Resilient distributed optimization algorithm against adversarial attacks,” IEEE Trans. Autom. Control, vol. 65, no. 10, pp. 4308–4315, Oct. 2020.

[28] W. M. Haddad and V. Chellaboina, Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach. Princeton, NJ, USA: Princeton Univ. Press, 2008.

[29] M. S. Stankovic, K. H. Johansson, and D. M. Stipanovic, “Distributed seeking of Nash equilibria with applications to mobile sensor networks,” IEEE Trans. Autom. Control, vol. 57, no. 4, pp. 904–919, Apr. 2012.
Dian Gadjov received the B.Sc. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, and the M.Sc. degree from the University of Toronto, Toronto, ON, where he is currently working toward the Ph.D. degree with the Systems Control Group. His research interests include (generalized) Nash equilibrium seeking over networks, distributed control, and optimization.

Lacra Pavel (Senior Member, IEEE) received the Diploma degree in engineering from the Technical University of Iasi, Iasi, Romania, and the Ph.D. degree in electrical engineering from Queen’s University, Kingston, ON, Canada. After a postdoctoral stage with the National Research Council and four years of working in the industry, in 2002, she joined the University of Toronto, Toronto, ON, where she is currently a Professor with the Department of Electrical and Computer Engineering. She is the author of the book titled Game Theory for Control of Optical Networks (Birkhäuser/Springer, 2012). Her research interests include game theory and distributed optimization in networks, with an emphasis on dynamics and control aspects.