Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

On Privatizing Equilibrium Computation in Aggregate Games over Networks

Shripad Gade **Anna Winnicki** **Subhonmesh Bose**

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, Urbana, IL 61801
{gade3, annaw5, boses}@illinois.edu

Abstract: We propose a distributed algorithm to compute an equilibrium in aggregate games where players communicate over a fixed undirected network. Our algorithm exploits correlated perturbation to obfuscate information shared over the network. We prove that our algorithm does not reveal private information of players to an honest-but-curious adversary who monitors several nodes in the network. In contrast with differential privacy based algorithms, our method does not sacrifice accuracy of equilibrium computation to provide privacy guarantees.

Keywords: Privacy, Nash Equilibrium Computation, Networked Aggregate Games.

1. INTRODUCTION

Aggregate games are non-cooperative games in which a player's payoff or cost depends on her own actions and the sum-total of the actions taken by other players. In a Cournot oligopoly for example, firms compete to supply a product in a market with a price-responsive demand with a goal to maximize profit. A firm's profit depends on her production cost as well as the market price, where the latter only depends on the aggregate quantity of the product offered in the market by all firms. Aggregate games are widely studied in the literature, e.g., see Novshek (1985); Jensen (2010). Multiple strategic interactions in practice admit an aggregate game model, e.g., Cournot competition models for wholesale electricity markets in Willems et al. (2009); Cai et al. (2019); Cherukuri and Cortés (2019), supply function competition in general economies see Jensen (2010), communication networks in Teng et al. (2019); Koskie and Gajic (2005) and common agency games in Martimort and Stole (2011). Aggregate games are often potential games and a pure-strategy Nash equilibrium can be guaranteed to exist. In this paper, we present an algorithm for networked players to compute such an equilibrium in a distributed fashion that maintains the privacy of players' cost structures.

Players in a networked game can only communicate with neighboring players in a communication graph. Distributed algorithms for computing Nash equilibrium in networked games have a rich literature, e.g., see Koshal et al. (2016); Salehisadaghiani and Pavel (2018); Ye and Hu (2017); Tatarenko et al. (2018); Parise et al. (2015). The obvious difficulty in computing equilibrium strategy arises due to the inability of a player to observe the aggregate decision. Naturally distributed Nash computation proceeds via iterative estimation of the aggregate decision followed by local payoff maximization (or cost minimization) with a given aggregate estimate. Koshal et al. (2016); Parise et al. (2015) exploit consensus based averaging, Koshal et al. (2016); Salehisadaghiani and Pavel (2018) explore gossip based averaging, and Tatarenko et al. (2018) employs gradient play along with acceleration for aggregate estimation over networks.

1.1 Our Contributions

Algorithms for equilibrium computation were not designed with privacy in mind. We show in Section 2.5, that an honest-but-curious adversary can compromise a few nodes in the network and observe the sequence of estimates to infer other players' payoff or cost structures for the algorithm in Koshal et al. (2016). In other words, information that allows distributed equilibrium computation can leak players' sensitive private information to adversaries.

Distributed equilibrium computation algorithms require aggregate estimates to update their own actions. Our proposed algorithm obfuscates local aggregate estimates before sharing them with neighbors. The obfuscation step involves players adding correlated perturbations to each outgoing aggregate estimate. The perturbations are designed such that they add to zero for each player. The received perturbed aggregate estimates are averaged by each player and used for updating strategy using local projected gradient descent.

Our main result (Theorem 1) reveals that obfuscation via correlated perturbations prevents an adversary from accurately learning cost structures provided the network satisfies appropriate connectivity conditions. Players converge to exact Nash equilibrium asymptotically. In other words, we simultaneously achieve both privacy and accuracy in distributed Nash computation in aggregate games. This is in sharp contrast to differentially private algorithms where trade-offs between accuracy and privacy guarantee are fundamental, e.g., see Han et al. (2016).

Simulations in Section 4 validate our results and corroborate our intuition that obfuscation slows down but does not impede the convergence of the algorithm.

⋆ This work is partially supported by a Siebel Energy Institute Grant and the Joan and Lalit Bahl Fellowship at Illinois. Authors would like to thank N. H. Vaidya for helpful comments.

2. EQUILIBRIUM COMPUTATION IN AGGREGATE GAMES AND THE LACK OF PRIVACY

We begin by introducing a networked aggregate game. We then present an adversary model and show that prior distributed equilibrium computation algorithms leak private information of players. This exposition motivates the development of privacy-preserving algorithms for equilibrium computation in the next section.

2.1 The Networked Aggregate Game Model

Consider a game with N players that can communicate over a fixed undirected network with reliable lossless links. Model this communication network by graph $\mathfrak{G}(\mathcal{V}, \mathcal{E})$, where each node in $\mathcal{V} := \{1, \dots, N\}$ denotes a player. Two players i and j can communicate with each other if and only if they share an edge in $\mathcal{E}$, denoted as $(i, j) \in \mathcal{E}$. Call $\mathcal{N}_i$ the set of neighbors of node i and $i \in \mathcal{N}_i$ by definition.

Player i can take actions in a convex compact set $\mathcal{X}_i \subseteq \mathbb{R}^d$, where $\mathbb{R}$ denotes the set of real numbers. Define $\overline{\mathcal{X}}$ as the Minkowski (set) sum of $\mathcal{X}_i$'s and

$$\overline{x} := \sum_{j=1}^{N} x_j$$

as the aggregate action of all players. For convenience, define $\overline{x}_{-i} := \sum_{j \neq i} x_i$. We assume that $\cap_{i=1}^{N} \mathcal{X}_i$ is nonempty. For an action profile $(x_1, \dots, x_N)$, player i incurs a cost that takes the form $f_i(x_i, \overline{x}) := f_i\left(x_i, x_i + \overline{x}_{-i}\right)$. This defines an aggregate game in that the actions of other players affect player i only through the sum of actions of all players, $\overline{x}$.

Each player $i \in \mathcal{V}$ thus seeks to solve

$$\begin{aligned} &\text{minimize} \quad f_i(x_i, x_i + \overline{x}_{-i}), \\ &\text{subject to} \quad x_i \in \mathcal{X}_i. \end{aligned} \tag{1}$$

For each $i \in \mathcal{V}$, assume that $f_i(x_i, y)$ is continuously differentiable in (x_i, y) over a domain that contains $\mathcal{X}_i \times \overline{\mathcal{X}}$. Furthermore, for each $i \in \mathcal{V}$, let $x_i \mapsto f_i(x_i, \overline{x})$ be convex over $\mathcal{X}_i$ and the gradient $\nabla_{x_i} f_i$ be uniformly $\overline{L}$-Lipschitz, i.e., $\exists\, \overline{L} > 0$ such that,

$$\|\nabla_{x_i} f_i(x_i, u) - \nabla_{x_i} f_i(x_i, u')\| \leq \overline{L} \|u - u'\|, \tag{2}$$

for all u, u' in $\overline{\mathcal{X}}$, x_i in $\mathcal{X}_i$. Throughout, $\|\cdot\|$ stands for the ℓ_2-norm of its argument. Define $\mathcal{X} := \times_{i=1}^{N} \mathcal{X}_i$ and the gradient map

$$\phi(x) := \begin{pmatrix} \nabla_{x_1} f_1(x_1, \overline{x}) \\ \vdots \\ \nabla_{x_N} f_N(x_N, \overline{x}) \end{pmatrix} \tag{3}$$

for $x := (x_1^\intercal, x_2^\intercal, \dots, x_N^\intercal)^\intercal \in \mathcal{X}$. Assume throughout that ϕ is strictly monotone over $\mathcal{X}$, i.e.,

$$[\phi(x) - \phi(x')]^\intercal (x - x') > 0, \tag{4}$$

for all $x, x' \in \mathcal{X}$ and $x \neq x'$. Denote this game in the sequel by $\mathsf{game}(\mathfrak{G}, \{f_i, \mathcal{X}_i\}_{i \in \mathcal{V}})$.

To provide a concrete example, consider the well-studied Nash-Cournot game (see Fudenberg and Tirole (1991)) among N suppliers competing to offer into a market for a single commodity where the price p varies with demand D as $p(D) := a - bD$. Supplier i offers to produce x_i amount of goods within its production capability modeled as $\mathcal{X}_i \subseteq \mathbb{R}_+$. Here $\mathbb{R}_+$ denotes the set of nonnegative real numbers. To produce x_i, supplier i incurs a cost of $c_i(x_i)$, where c_i is increasing, convex and differentiable. Each supplier seeks to maximize her profit, or equivalently, minimize her loss. The loss of supplier i is

$$f_i(x_i, \overline{x}) = c_i(x_i) - x_i p(\overline{x}) = c_i(x_i) - x_i(a - b\overline{x}).$$

2.2 Equilibrium Definition and Existence

An action profile $(x_1^*, \dots x_N^*)$ defines a Nash equilibrium of $\mathsf{game}(\mathfrak{G}, \{f_i, \mathcal{X}_i\}_{i \in \mathcal{V}})$ in pure strategies, if

$$f_i\left(x_i^*, x_i^* + \overline{x}_{-i}^*\right) \leq f_i\left(x_i, x_i + \overline{x}_{-i}^*\right),$$

for all $x_i \in \mathcal{X}_i$ and $i \in \mathcal{V}$.

The networked aggregate game, as described above, always admits a unique pure strategy Nash equilibrium. See Theorem 2.2.3 in Facchinei and Pang (2007) for details. Given that an equilibrium always exists, prior literature has studied distributed algorithms for players to compute such an equilibrium.

2.3 Prior Algorithms for Distributed Nash Computation

We now describe the distributed algorithm in Koshal et al. (2016) for equilibrium computation of $\mathsf{game}(\mathfrak{G}, \{f_i, \mathcal{X}_i\}_{i \in \mathcal{V}})$. In Section 2.5, we demonstrate that adversarial players can infer private information about cost structures f_i's from observing a subset of the variables during equilibrium computation using that algorithm. While we only study the algorithm in Koshal et al. (2016), our analysis can be extended to those presented in Salehisadaghiani and Pavel (2018); Ye and Hu (2017); Tatarenko et al. (2018); Parise et al. (2015).

Recall that players in $\mathsf{game}(\mathfrak{G}, \{f_i, \mathcal{X}_i\}_{i \in \mathcal{V}})$ do not have access to the aggregate decision. To allow equilibrium computation, let players at iteration k maintain estimates of the aggregate decision $\overline{x}$ as $v_1^k, \dots, v_N^k$, initialized as, $v_i^0 = x_i^0$ for each player i. At discrete time steps $k \geq 0$, each player transmits her own estimate of the aggregate decision to its neighbors and updates her own action as,

$$\widehat{v}_i^k = \sum_{j=1}^{N} W_{ij} v_j^k, \tag{5a}$$

$$x_i^{k+1} = \mathrm{proj}_{\mathcal{X}_i}\left[x_i^k - \alpha^k \nabla_{x_i} f_i(x_i^k, N\widehat{v}_i^k)\right], \tag{5b}$$

$$v_i^{k+1} = \widehat{v}_i^k + x_i^{k+1} - x_i^k. \tag{5c}$$

Here, $\mathrm{proj}_{\mathcal{X}_i}$ stands for projection on $\mathcal{X}_i$, and α^k is a common learning rate of all players.

The algorithm has three steps. First, player i computes a weighted average of the estimates of the aggregate received from its neighbors in (5a), where W is a symmetric doubly-stochastic weighting matrix. The sparsity pattern of the matrix follows that of graph $\mathfrak{G}$, i.e.,

$$W_{ij} \neq 0 \iff (i, j) \in \mathcal{E}.$$

Second, player i performs a projected gradient update in (5b) utilizing the weighted average of local aggregate decision $\widehat{v}_i^k$ in lieu of the true aggregate decision $\overline{x}$. Finally, she updates her own estimate of aggregate average in (5c) based on her local decision x_i^k and its update x_i^{k+1}.

2.4 Adversary Model and Privacy Definition

Consider an adversary A that compromises the players in $\mathcal{A} \subseteq \mathcal{V}$. A is equipped with unbounded storage and computational capabilities, and has access to all information stored, processed locally and communicated to any compromised players at all times. We define adversary model using the information available to A.

(A) For a compromised node $i \in \mathcal{A}$, A knows all local information f_i, x_i^k, v_i^k, $\widehat{v}_i^k$ and information received from neighbors of i i.e., v_j^k for $j \in \mathcal{N}_i$ at each $k \geq 0$.
(B) A knows the algorithm for equilibrium computation and its parameters $\{\alpha^k\}$ and W.
(C) A observes aggregate decision $\overline{x}^k$ at each k.

What does A seek to infer? The dependency of a player's cost on her own actions encodes private information. In the Cournot competition example, this dependency is precisely supplier i's production cost – information that is business sensitive. A seeks to exploit information sequence observed from compromised players to infer private information of other players. Intuitively, privacy implies inability of A to infer private cost functions.

Denote the set of non-adversarial nodes by $\mathcal{A}^c := \mathcal{V} \setminus \mathcal{A}$. Call $\mathfrak{G}(\mathcal{A}^c)$ the restriction of $\mathfrak{G}$ to $\mathcal{A}^c$ obtained by deleting the adversarial nodes. See Figure 1 for an illustration. For this example, A monitors all variables and parameters pertaining to player 5, but seeks to infer the functions $f_1, \ldots, f_4$.

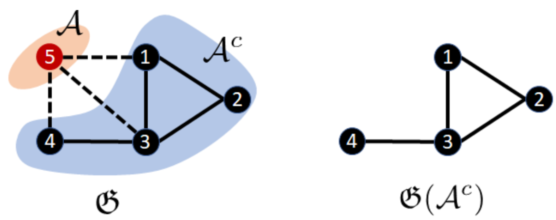

Fig. 1. Illustration of $\mathfrak{G}$ and $\mathfrak{G}(\mathcal{A}^c)$. Here, $\mathcal{A} = \{5\}$ and $\mathcal{A}^c = \{1, 2, 3, 4\}$.

Let Π denote the set of all permutations over all non-adversarial nodes in $\mathcal{A}^c$. Define the collection of games

$$\mathcal{F} := \left\{ \mathsf{game}(\mathfrak{G}, \{f_{\pi(i)}, \mathcal{X}_{\pi(i)}\}_{i \in \mathcal{V}}) \;\middle|\; \pi \in \Pi \right\}.$$

Thus, $\mathcal{F}$ comprises the games where the cost functions and strategy sets of non-adversarial players are permuted. All games in $\mathcal{F}$ have the same aggregate strategy $\overline{x}^*$ at Nash equilibrium. Next, we utilize $\mathcal{F}$ to define privacy.

Definition 1. (Privacy). Consider a distributed algorithm to compute the Nash equilibrium of $\mathsf{game}(\mathfrak{G}, \{f_i, \mathcal{X}_i\}_{i \in \mathcal{V}})$. If execution observed by adversary A is consistent with all games in $\mathcal{F}$, then the algorithm is private.

We define privacy as the inability of A to distinguish between games in $\mathcal{F}$. Even if A knew all possible costs exactly–which is a tall order–our privacy definition implies that A cannot associate such costs to specific players.

2.5 Privacy Breach in Algorithm (5)

Consider a Cournot competition among 5 players connected according to $\mathfrak{G}$ in Figure 1, where A has compromised player 5. Assume that the equilibrium of the game lies in the interior of each player's strategy set. Recall that A stores observed information at each k and processes it to infer private cost information $c_i(x_i)$. We argue how A can compute cost functions $c_1(\cdot), \ldots, c_4(\cdot)$ up to a constant.

We first show privacy breach for player 4. A observes $\{v_1^k, v_3^k, v_4^k, v_5^k\}$ at each $k \geq 0$. A uses v_3^k, v_4^k, v_5^k and W to compute $\widehat{v}_4^k$ using (5a). Moreover, A uses (5c) to compute,

$$x_4^{k+1} - x_4^k = v_4^{k+1} - \widehat{v}_4^k.$$

For large enough k, the step-size α^k is small enough to ensure,

$$\mathrm{proj}_{\mathcal{X}_i}\left[x_i^k - \alpha^k \nabla_{x_i} f_i(x_i^k, N\widehat{v}_i^k)\right] \approx x_i^k - \alpha^k \nabla_{x_i} f_i(x_i^k, N\widehat{v}_i^k).$$

At such large k, A uses (5b) along with $(x_4^{k+1} - x_4^k)$ and α^k to calculate $\nabla_{x_4} f_4(x_4^k, N\widehat{v}_4^k)$.

A uses information about strucutre of loss function i.e. $f_4(x_4, \overline{x}) = c_4(x_4) - x_4(a - b\overline{x})$, along with $\nabla_{x_4} f_4(x_4^k, N\widehat{v}_4^k)$, $\widehat{v}_4^k$, $\overline{x}^k$ and game parameters a, b to learn $c_4'(x_4^k)$. Several observations of $(x_4^k, c_4'(x_4^k))$ allows A to learn the private cost c_4 upto a constant.

We showed that privacy breach for player 4, the same analysis can be used for players 1, 2 and 3 with an additional step. A observes $\overline{x}^k$, which tracks $\frac{1}{N}\sum_i v_i^k$ (Lemma 2 in Koshal et al. (2016)). A computes

$$v_2^k = N\overline{x}^k - (v_1^k + v_3^k + v_4^k + v_5^k).$$

Since $\{v_2^k\}$ is available for each $k \geq 0$, A uses same process as above to show privacy breach for players 1, 2 and 3.

For algorithm (5), A uncovers all private cost functions $c_i(\cdot)$ for an example aggregate game. Next, we design an algorithm that protects privacy of players' private information in the sense of Definition 1 against A.

3. OUR ALGORITHM AND ITS PROPERTIES

We propose and analyze Algorithm 1 that computes Nash equilibrium of $\mathsf{game}(\mathfrak{G}, \{f_i, \mathcal{X}_i\}_{i \in \mathcal{V}})$ in a distributed fashion. The main result (Theorem 1) shows that the algorithm asymptotically converges to the equilibrium. Attempts by A to recover each player's cost structure, however, remain unsuccessful.

The key idea behind our design is the injection of correlated noise perturbations in the exchange of local estimates of the aggregate decision. Different neighbors of player i receive different estimates of the aggregate decision. The perturbations added by any player i add to zero. While A may still infer the true aggregate decision, the protocol does not allow him to correctly infer the players' iterates or the gradients of their costs with respect to their own actions. Our assumption on network connectivity requires $\mathcal{G}(\mathcal{A}^c)$ be connected and not be bipartite. Under these conditions A cannot monitor all outgoing communication channels from any player. We further show that one can design noises in a way that A's observations are consistent with all games in $\mathcal{F}$, making it impossible for him to uncover cost for any specific player.

Throughout, assume that W is a doubly stochastic that follows the sparsity pattern of $\mathfrak{G}$. Further, assume that all non-diagonal, non-zero entries of W are identically $\delta < \frac{1}{N-1}$.

Algorithm 1 Private Distributed Nash Computation

Input: Player i knows $f_i(x_i, \overline{x})$, $\mathcal{X}_i$, and δ. Consider a non-increasing non-negative sequence $\boldsymbol{\alpha}$ that satisfies

$$\sum_{k=1}^{\infty} \alpha^k = \infty \text{ and } \sum_{k=1}^{\infty} [\alpha^k]^2 < \infty. \tag{6}$$

Initialize: For $i \in \mathcal{V}$, $v_i^0 = x_i^0 = \chi \in \cap_i \mathcal{X}_i$.

For $k \geq 0$, players $i \in \mathcal{V}$ execute in parallel:

1: Construct $|\mathcal{N}_i|$ random numbers $\{r_{ij}^k\}$, satisfying

$$r_{ii}^k = 0 \text{ and } \sum_{j \in \mathcal{N}_i} r_{ij}^k = 0. \tag{7}$$

2: Send obfuscated aggregate estimates v_{ij}^k to $j \in \mathcal{N}_i$, where

$$v_{ij}^k = v_i^k + \alpha^k r_{ij}^k. \tag{8}$$

3: Compute weighted average of received estimates v_{ji}^k as

$$\widehat{v}_i^k = \sum_{j=1}^{N} W_{ij} v_{ji}^k. \tag{9}$$

4: Perform a projected gradient descent step as

$$x_i^{k+1} = \text{proj}_{\mathcal{X}_i}[x_i^k - \alpha^k \nabla_{x_i} f_i(x_i^k, N\widehat{v}_i^k)]. \tag{10}$$

5: Update local aggregate estimate as

$$v_i^{k+1} = \widehat{v}_i^k + x_i^{k+1} - x_i^k. \tag{11}$$

At each time k, player i generates correlated random numbers $\{r_{ij}^k\}$ satisfying $r_{ii}^k = 0$ and $\sum_{j \in \mathcal{N}_i} r_{ij}^k = 0$. Player i then adds $\alpha^k r_{ij}^k$ to v_i^k to generate v_{ij}^k, the estimate sent by player i to player j, according to (8). Let $\boldsymbol{r}$ denote the collection of r's for all players across time. Call $\boldsymbol{r}$ the obfuscation sequence.

Each node i computes weighted average of received aggregate estimates v_{ji}^k to construct its own estimate aggregate decision $N\widehat{v}_i^k$, following (9). Players perform projected gradient descent using local decision estimate x_i^k, gradient of cost function $\nabla_{x_i} f_i(x_i^k, N\widehat{v}_i^k)$, and non-summable, square-summable step size α^k (see (6)) to arrive at an improved local decision estimate x_i^{k+1} using (10). Players then update their local aggregate estimate using the change in local decision estimate $x_i^{k+1} - x_i^k$ per (11). The properties of our algorithm are summarized in the next result. A brief proof sketch is included in Section 5.

Theorem 1. Consider a networked aggregate game defined as $\mathsf{game}(\mathfrak{G}, \{f_i, \mathcal{X}_i\}_{i \in \mathcal{V}})$. If $\mathfrak{G}(\mathcal{A}^c)$ is connected and not bipartite, then Algorithm 1 is private. Moreover, if the obfuscation sequence is bounded, then Algorithm 1 asymptotically converges to a Nash equilibrium of the game.

The convergence properties largely mimic that of distributed descent algorithms for equilibrium computation. The locally balanced and bounded nature of the designed noise together with decaying step-sizes ultimately drown the effect of the noise. Computing balanced yet bounded perturbations can be achieved using secure multiparty computation protocols described in Gade and Vaidya (2016, 2018b); Abbe et al. (2012). Our assumption on $\mathfrak{G}(\mathcal{A}^c)$ is such that given two games $F, \tilde{F}$ from $\mathcal{F}$ and an obfuscation sequence $\boldsymbol{r}$, we are able to design a different obfuscation sequence $\tilde{\boldsymbol{r}}$, such that the execution of F perturbed with $\boldsymbol{r}$ generates identical observables as $\tilde{F}$ perturbed with $\tilde{\boldsymbol{r}}$. The connectivity among non-adversarial players in $\mathcal{A}^c$ is key to the success of our algorithm design. Convergence speed depends on the size of the perturbations. We investigate this link experimentally in Section 4, but leave analytical characterization of this relationship for future work. In what follows, we compare our algorithm and its properties to other protocols for privacy preservation.

Comparison with Differentially Private Algorithms: Differentially private algorithms for computing Nash equilibrium of potential games have been studied in Dong et al. (2015); Cummings et al. (2015). The algorithm in Dong et al. (2015) executes a differentially private distributed mirror-descent algorithm to optimize the potential function. Experiments reveal that a trade-off arises between accuracy and privacy parameters, i.e., the more privacy one seeks, the less accurate the final output of the algorithm becomes. Such a tradeoff is a hallmark of differentially private algorithms, e.g., see Han et al. (2016). Our algorithm on the other hand does not suffer from that limitation. Notice that our definition of privacy is binary in nature. That is, an algorithm for equilibrium computation can either be private or non-private. We aim to explore properties of our algorithmic architecture with notions of privacy that allow for a degree of privacy and compare them with differentially private algorithms.

Comparison with Cryptographic Methods: Authors in Lu and Zhu (2015) use secure multiparty computation to compute Nash equilibrium. Such an approach guarantees privacy in an information theoretic sense. This protocol provides privacy guarantees along with accuracy, similar to our algorithmic framework. However, cryptographic protocols are typically computationally expensive for large problems (see Section V in Zhang et al. (2019)), and are often difficult to implement in distributed settings.

Comparison with Non-observability based Methods: The authors in Shakarami et al. (2019); Monshizadeh and Tabuada (2019) use the "plausible deniability" principle, which implies that private information cannot be reconstructed by an adversary. Similar to our approach, these methods also preserve accuracy and privacy simultaneously. However, our privacy result guarantees same adversarial observations under network-permutation of payoff functions, whereas the result in Shakarami et al. (2019) protects the initial states of the system to make reconstruction difficult.

Comparison to Private Distributed Optimization: Our earlier work in Gade and Vaidya (2018b) has motivated the design of Algorithm 1. While our prior work seeks privacy-preserving distributed protocols to cooperatively solve optimization problems, the current paper focuses on non-cooperative games. Protocols in Gade and Vaidya (2018b) advocate use of perturbations that cancel over the network. Such a design is not appropriate for networked games for two reasons. First, players must agree on noise design, a premise that requires cooperation. Second, per-

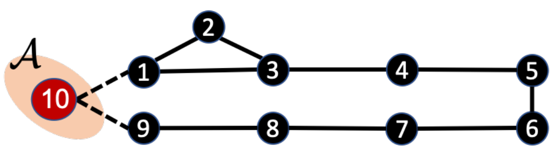

Fig. 2. Communication network for Cournot network example on $N = 10$ players.

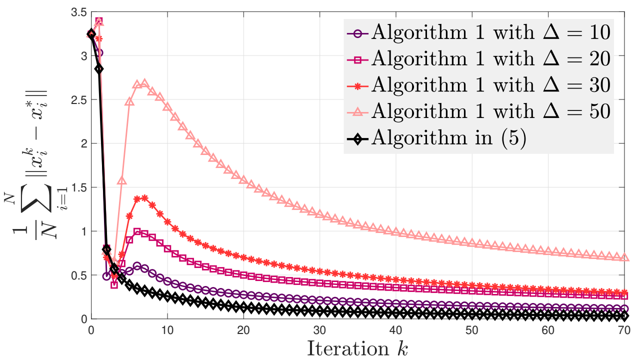

Fig. 3. Iterates generated by Algorithm 1 versus the Algorithm in (5) for $\Delta = \{10, 20, 30, 50\}$.

turbing local functions f_i's, even if the changes cancel in aggregate, can alter the equilibrium of the game.

Privacy in Client-Server architecture: This work considers players communicating over a peer-to-peer network. However, engineered distributed systems often have a client-server (federated) architecture. Presence of a central server entity allows for easy aggregate computation. However, privacy is sacrificed if the parameter server is adversarial. We have investigated private distribution optimization in this architecture in Gade and Vaidya (2018a), where, we use multiple servers instead of one, a subset of which can be adversarial. We believe our algorithm design and analysis in Gade and Vaidya (2018a) can be extended to deal with private equilibrium computation for aggregate games in client-server framework.

4. A NUMERICAL EXPERIMENT

Consider a Cournot competition with $N = 10$ players over $\mathfrak{G}$ described in Figure 2. Player i's cost is given by

$$c_i(x_i) = \zeta_{i,2}x_i^2 + \zeta_{i,1}x_i.$$

The cost coefficients are drawn randomly from

$$\zeta_{i,2} \sim \mathsf{unif}[0, 1/2],\ \ \zeta_{i,1} \sim \mathsf{unif}[0, 1]$$

for each i. The strategy sets are identically $\mathcal{X}_i = [0, 5]$ for each i. Choose $\delta = \frac{1}{10}$ that parameterizes the matrix W. Let the price vary with demand D as

$$p(D) = 6 - \frac{1}{10}D.$$

We initialize the algorithm with $x^0 = 1$ identically for all players. We use secure multi-party computing technique in Gade and Vaidya (2018b) to design obfuscation sequence $\boldsymbol{r}$ that satisfies (7) and

$$|r_{ij}^k| \leq \Delta.$$

The trajectory of the average distance of x_i^k's from x_i^* across players with $\alpha_k := (k+1)^{0.51}$ is shown in Figure 3. Our algorithm converges to the equilibrium similar to the non-private algorithm in (5). However, its convergence is slower as seen in Figure 3. The slowdown is especially pronounced for large Δ's and is an artifact of perturbations added by players to obfuscate information from the adversary. Thus, our algorithm design achieves privacy and asymptotic convergence to equilibrium, but sacrifices speed of convergence. An analytical characterization of the slowdown defines an interesting direction for future work.

5. PROOF SKETCH OF THEOREM 1

We provide a brief proof sketch of privacy and correctness results (Theorem 1). Detailed proofs are included in the longer version of this paper, Gade et al. (2019).

5.1 *Proving Algorithm 1 is Private*

We consider two games, F and $\tilde{F}$, obtained by two non-adversarial nodes switching their private information viz. cost function and local strategy set. These two problems belong to $\mathcal{F}$. We show that, under Algorithm 1, the execution observed by the adversary is the same for both games F and $\tilde{F}$. Privacy claim follows from Definition 1. In what follows, we sketch this construction and proof.

Recall, $\mathcal{A}^c$ represents the set of non-adversarial nodes. Let I, J be any two players in $\mathcal{A}^c$ and

$$F := (f_i, \mathcal{X}_i)_{i\in\mathcal{V}}, \quad \tilde{F} := (\tilde{f}_i, \tilde{\mathcal{X}}_i)_{i\in\mathcal{V}},$$

be two games in $\mathcal{F}$ such that $\tilde{F}$ is identical to F, except that costs and strategy sets of players I and J are switched:

$$\tilde{f}_I = f_J,\ \tilde{f}_J = f_I,\ \tilde{\mathcal{X}}_I = \mathcal{X}_J,\ \tilde{\mathcal{X}}_J = \mathcal{X}_I.$$

For convenience, define $\pi : \mathcal{V} \to \mathcal{V}$ as the permutation that encodes the switch, i.e.,

$$\pi(I) = J,\ \pi(J) = I,\ \text{and}\ \pi(i) = i \text{ for all } i \neq I, J.$$

Note, an arbitrary permutation over $\mathcal{A}^c$ is equivalent to a composition of a sequence of switches among two players in $\mathcal{A}^c$. Consequently, our proof for a simple switch is sufficient to show that algorithm execution on games in $\mathcal{F}$ can be made to appear identical from A's standpoint, proving the privacy of Algorithm 1.

Consider the execution of Algorithm 1 on F, given by

$$\mathsf{E}(F, \boldsymbol{r}, \chi) := \{(x_i^k, v_i^k, \hat{v}_i^k) \text{ for } i \in \mathcal{V}, k \geq 0\},$$

with obfuscation sequence $\boldsymbol{r}$ used in (8), initialized with $\chi \in \cap_{i=1}^N \mathcal{X}_i$. We prove that there exists an obfuscation sequence $\tilde{\boldsymbol{r}}$ such that execution $\mathsf{E}(\tilde{F}, \tilde{\boldsymbol{r}}, \chi)$ of Algorithm 1 on $\tilde{F}$ with $\tilde{\boldsymbol{r}}$ starting from χ, is identical to $\mathsf{E}(F, \boldsymbol{r}, \chi)$, from A's perspective.

Adversary observes $\{x_j^k, v_j^k, \hat{v}_j^k\}$ for all $j \in \mathcal{A}$ at each $k \geq 0$. Forcing executions $\mathsf{E}(F, \boldsymbol{r}, \chi)$ and $\mathsf{E}(\tilde{F}, \tilde{\boldsymbol{r}}, \chi)$ to be equivalent, based on adversarial observations, leads us to linear equations in $\tilde{\boldsymbol{r}}$. We show that the connected and non-bipartite nature of residual graph $\mathfrak{G}(\mathcal{A}^c)$ is sufficient for the system of linear equations (in $\tilde{\boldsymbol{r}}$) to have at least one solution. Equivalently, if $\mathfrak{G}(\mathcal{A}^c)$ is connected and not bipartite, then there exists $\tilde{\boldsymbol{r}}$ that ensures $\mathsf{E}(\tilde{F}, \tilde{\boldsymbol{r}}, \chi)$ and $\mathsf{E}(F, \boldsymbol{r}, \chi)$ appear identical to A. This results in A being unable to differentiate between games F, $\tilde{F}$ in $\mathcal{F}$, allowing us to claim privacy.

5.2 Proving Algorithm 1 Converges to Nash Equilibrium

The convergence proof follows a similar path as in Koshal et al. (2016). The proof involves generating a dissipation inequality similar to Koshal et al. (2016), albeit with additional terms. The additional terms are a consequence of the obfuscation sequence and its effects on the (strategy) iterates. We use the doubly stochastic nature of W, property of step-sizes α_k from (6), and two properties of obfuscation sequence – boundedness of $\boldsymbol{r}$ and local balancedness property from (7) – to show that the additional terms are ℓ_1-summable. This along with convergence of non-negative almost supermartingales (Theorem 1 in Robbins and Siegmund (1985)) and strict monotonicity of ϕ allows us to demonstrate asymptotic convergence of the iterates to the Nash Equilibrium.

6. CONCLUSIONS

In this paper, we considered aggregate games played by agents that communicate over a network, each with private information. We showed that distributed algorithms for equilibrium computation in the literature are not designed with privacy requirements in mind, and consequently leak private information about players against honest-but-curious adversaries. Our proposed algorithm for NE computation exploits correlated perturbations to obfuscate aggregate estimates shared over the network. The algorithm asymptotically converges to the Nash Equilibrium. If the graph connecting non-adversarial players is connected and not bipartite, we show that our algorithm protects private information of non-adversarial players.

REFERENCES

Abbe, E.A., Khandani, A.E., and Lo, A.W. (2012). Privacy-preserving methods for sharing financial risk exposures. *The American Economic Review*, 102(3), 65–70.

Cai, D., Bose, S., and Wierman, A. (2019). On the role of a market maker in networked cournot competition. *Mathematics of Operations Research*.

Cherukuri, A. and Cortés, J. (2019). Iterative bidding in electricity markets: rationality and robustness. *IEEE Transactions on Network Science and Engineering*.

Cummings, R., Kearns, M., Roth, A., and Wu, Z.S. (2015). Privacy and truthful equilibrium selection for aggregative games. In *International Conference on Web and Internet Economics*, 286–299. Springer.

Dong, R., Krichene, W., Bayen, A.M., and Sastry, S.S. (2015). Differential privacy of populations in routing games. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 2798–2803. IEEE.

Facchinei, F. and Pang, J.S. (2007). *Finite-dimensional variational inequalities and complementarity problems.* Springer Science & Business Media.

Fudenberg, D. and Tirole, J. (1991). Game theory, 1991. *Cambridge, Massachusetts*, 393(12), 80.

Gade, S. and Vaidya, N.H. (2016). Private learning on networks. *arXiv preprint arXiv:1612.05236.*

Gade, S. and Vaidya, N.H. (2018a). Privacy-preserving distributed learning via obfuscated stochastic gradients. In *2018 IEEE Conference on Decision and Control (CDC)*, 184–191. IEEE.

Gade, S. and Vaidya, N.H. (2018b). Private optimization on networks. In *2018 Annual American Control Conference (ACC)*, 1402–1409. IEEE.

Gade, S., Winnicki, A., and Bose, S. (2019). On privatizing equilibrium computation in aggregate games over networks. *arXiv preprint arXiv:1912.06296.*

Han, S., Topcu, U., and Pappas, G.J. (2016). Differentially private distributed constrained optimization. *IEEE Transactions on Automatic Control*, PP(99), 1–1. doi: 10.1109/TAC.2016.2541298.

Jensen, M.K. (2010). Aggregative games and best-reply potentials. *Economic theory*, 43(1), 45–66.

Koshal, J., Nedić, A., and Shanbhag, U.V. (2016). Distributed algorithms for aggregative games on graphs. *Operations Research*, 64(3), 680–704.

Koskie, S. and Gajic, Z. (2005). A nash game algorithm for sir-based power control in 3g wireless cdma networks. *IEEE/ACM Transactions on Networking (TON)*, 13(5), 1017–1026.

Lu, Y. and Zhu, M. (2015). Game-theoretic distributed control with information-theoretic security guarantees. *IFAC-PapersOnLine*, 48(22), 264–269.

Martimort, D. and Stole, L. (2011). Aggregate representations of aggregate games.

Monshizadeh, N. and Tabuada, P. (2019). Plausible deniability as a notion of privacy. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 1710–1715. IEEE.

Novshek, W. (1985). On the existence of cournot equilibrium. *The Review of Economic Studies*, 52(1), 85–98.

Parise, F., Gentile, B., Grammatico, S., and Lygeros, J. (2015). Network aggregative games: Distributed convergence to nash equilibria. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 2295–2300. IEEE.

Robbins, H. and Siegmund, D. (1985). A convergence theorem for non negative almost supermartingales and some applications. In *Herbert Robbins Selected Papers*, 111–135. Springer.

Salehisadaghiani, F. and Pavel, L. (2018). Distributed nash equilibrium seeking in networked graphical games. *Automatica*, 87, 17–24.

Shakarami, M., De Persis, C., and Monshizadeh, N. (2019). Privacy and robustness guarantees in distributed dynamics for aggregative games. *arXiv preprint arXiv:1910.13928.*

Tatarenko, T., Shi, W., and Nedić, A. (2018). Accelerated gradient play algorithm for distributed nash equilibrium seeking. In *2018 IEEE Conference on Decision and Control (CDC)*, 3561–3566. IEEE.

Teng, Z.j., Xie, L.y., Chen, H.l., Teng, L.x., and Li, H.b. (2019). Application research of game theory in cognitive radio spectrum allocation. *Wireless Networks*, 1–12.

Willems, B., Rumiantseva, I., and Weigt, H. (2009). Cournot versus supply functions: What does the data tell us? *Energy Economics*, 31(1), 38–47.

Ye, M. and Hu, G. (2017). Distributed nash equilibrium seeking by a consensus based approach. *IEEE Transactions on Automatic Control*, 62(9), 4811–4818.

Zhang, C., Ahmad, M., and Wang, Y. (2019). Admm based privacy-preserving decentralized optimization. *IEEE Transactions on Information Forensics and Security*, 14(3), 565–580.