

# Learning Rationality in Potential Games

Stefan Clarke<sup>1</sup>, Gabriele Dragotto<sup>1</sup>, Jaime Fernández Fisac<sup>2</sup>, and Bartolomeo Stellato<sup>1</sup>

<sup>1</sup> Department of Operations Research and Financial Engineering, Princeton University, U.S.A.

<sup>2</sup> Department of Electrical and Computer Engineering, Princeton University, U.S.A.

**Abstract**—We propose a stochastic first-order algorithm to learn the *rationality* parameters of simultaneous and non-cooperative potential games, *i.e.*, the parameters of the agents' optimization problems. Our technique combines an active-set step that enforces that the agents play at a Nash equilibrium and an implicit-differentiation step to update the estimates of the rationality parameters. We detail the convergence properties of our algorithm and perform numerical experiments on Cournot and congestion games. In practice, we show that our algorithm effectively finds high-quality solutions with minimal out-of-sample loss and scales to large datasets.

## I. INTRODUCTION

Decision-making is rarely an individual task; on the contrary, it often involves several self-driven interacting agents. Non-cooperative game theory provides the ideal playground to model the outcome of multi-agent decision-making. Among these outcomes, the most popular one for *non-cooperative* and *simultaneous* games is arguably the Nash equilibrium [1]. Intuitively, whenever the game's agents play Nash equilibrium strategies, no agent has the incentive to deviate to another strategy, and the multi-agent system is stable. Monderer and Shapley [2] proved that Nash equilibria are guaranteed to exist in *potential* games, a large subclass of non-cooperative and simultaneous games; specifically, if the *rationality parameters* of a potential game, *i.e.*, the parameters describing the agents' payoffs and strategy sets, are common information and observable, there always exists a Nash equilibrium that is an extremum of a so-called *potential function*. Several games with practical applications can be modeled as *potential games*, for example, congestion games, Cournot games [2], and certain urban driving games [3].

However, the agents' self-driven behavior frequently conflicts with the greater societal goals; in such cases, external regulators (*e.g.*, governments) can intervene in the agents' interaction via incentives, laws, and regulations. For instance, the transportation company of a large city adds or removes subway rides starting from its understanding of the commuters' preferences. Therefore, accurately estimating the rationality parameters as a proxy of the agents' behavior is arguably the most critical ingredient for deploying effective regulatory interventions. In the specific case of potential games, if the external regulator can directly observe the rationality parameters, any extrema of the potential function is a Nash equilibrium. Assuming that the regulator can directly observe such rationality parameters is, however, rather unrealistic; in several contexts, these parameters are often unknown and not directly observable. Although the rationality parameters are seldom observable, outcomes such as Nash equilibria are often observable through data. The observability of outcomes, as opposed to the observability of the rationality parameters, is the starting assumption of the techniques we present here. In this paper, we propose an algorithm that learns the rationality parameters starting from some data containing the outcomes (*i.e.*, Nash equilibria) of the past agents' interactions. Our algorithm enables external regulators to learn the agents' rationality parameters from data to deploy effective regulatory interventions. We assume that each agent solves a convex quadratic optimization problem parametrized in some unknown *rationality* parameters

and some *context* parameters (*e.g.*, some observable environmental conditions), and, further, that there exists a convex quadratic potential function in the agents' variables.

The majority of the learning literature focused on the problem of learning the optimal strategy from the perspective of the agents [4, 5, 6, 7]. In this paper, we study the less explored topic of learning the rationality parameters from the perspective of an external regulator, as in [8, 9, 10, 11, 12]. In contrast with the previous literature, our algorithm: (a.) learns rationality parameters that induce *exact* and *exact generalized* Nash equilibria, instead of approximate equilibria or Nash equilibria [9, 10, 11], and (b.) is an *anytime* algorithm, *i.e.*, the intermediate estimates always induce exact Nash equilibria, as opposed to [8, 9, 10, 11, 12], and (c.) learns both the agents' utilities and constraints, as opposed to only utilities [8], and (d.) scales to high-dimensional datasets. We summarize our contributions as follows:

- 1) We propose a stochastic *active-set-based* first-order algorithm to learn the rationality parameters of the agents of a potential game given a dataset of past observed equilibria and context parameters.
- 2) We provide a detailed analysis of the asymptotic convergence properties of our algorithm.
- 3) We showcase the effectiveness and efficiency of our algorithm via computational experiments on Cournot and congestion games.

The code associated with this work is available at [https://github.com/stellatogrp/learning\\_rationality\\_in\\_potential\\_games](https://github.com/stellatogrp/learning_rationality_in_potential_games).

## II. RELATED WORK

### A. Learning in games

The majority of the learning literature focuses on the problem of learning optimal strategies from the perspective of the agents [4, 5, 6, 7]. These approaches are useful to determine suitable agent strategies, yet it is hard to use these techniques to determine how to correctly *intervene* in the game to achieve a specific purpose. In this paper, we study the less explored topic of learning the parameters of the game, *i.e.*, affecting the utilities and the strategies of the agents, from the perspective of an external observer [8]. In contrast with [13, 14, 15], our approach does not employ normal or extensive-form formulations. Instead, we let each agent decide their strategy by solving a parametrized optimization problem, as in [16, 17]. Furthermore, compared to [8], we allow agents to have individual constraints parametrized in both rationality and context parameters.

### B. Inverse equilibrium

Our work is closely related to the *inverse equilibrium problem*, *i.e.*, the problem using data to estimate the parameters affecting the game. Several works tackle these problems using variational inequalities [11] and implicit neural networks [10], with applications in reinforcement learning [18], decision making [19], and motion planning [20, 21]. However, these works often do not scale up to large datasets and do not provide anytime algorithms. In contrast, our

method models the inverse learning task as a (nonconvex quadratic) optimization problem, and, in this sense, it is closer to [22] in the context of learning the parameters of convex optimization models. Similarly to [23], we employ a complementarity problem for our learning task; however, our algorithm enforces exact complementarity at each iteration whereas [23] incorporates it as a violation-based penalty function.

### C. Stochastic Methods

Several authors employed stochastic gradient descent-type methods to estimate the parameters in games. Geiger and Straehle [24] predict rationality parameters in urban driving games using a neural network to anticipate the future trajectories of vehicles. They a neural network, the “preference revelation net”, to map car trajectories to rationality parameters for each driver. The authors also optimize the neural network using the implicit function theorem. In particular, they train a neural network to predict the “rationality” parameters for a given class of trajectory games, specifically driving games. In contrast, in our work, we directly estimate the game parameters without relying on the predictions from a neural network, and we consider quadratic potential games. Similarly, [19] model an inverse-equilibrium task in motion planning problems and focus on quantal response equilibrium, *i.e.*, a different equilibrium concept. In contrast to the previous works, we model the learning task as an optimization problem with complementarity constraints and consider Nash equilibria. Several works in bilevel optimization, *e.g.*, [25, 26], combine stochastic methods and implicit differentiation to solve bilevel (*i.e.*, nested) optimization problems; while these techniques do not focus on simultaneous games, they may be extended to estimate the rationality parameters in this setting.

## III. PROBLEM FORMULATION

We consider a class of potential games among  $n$  agents solving optimization problems as in Definition III.1. We assume there are known and observable context parameters  $\mu$  and some unknown rationality parameters  $\theta$ , which we seek to infer.

**Definition III.1** (Mathematical programming game). *Let  $\theta \in \mathbf{R}^p$  be the vector of rationality parameters, and  $\mu \in \mathbf{R}^k$  be the vector of context parameters. We consider a non-cooperative and simultaneous game among  $n$  agents such that each agent  $i = 1, \dots, n$  solves the (parametrized) optimization problem*

$$\begin{aligned} & \underset{x_i \in \mathbf{R}^m}{\text{minimize}} && u_i(x_i; x_{-i}, \theta, \mu) \\ & \text{subject to} && B_i(\theta, \mu)x_i + D_i(\theta, \mu)x_{-i} \leq b_i(\theta, \mu), \end{aligned} \quad (1)$$

where  $x \in \mathbf{R}^{mn}$  is the strategy profile made of the decisions of all agents, including other agents’ decisions  $x_{-i}$  and the decisions  $x_i$  of  $i$ ;  $B_i(\theta, \mu) \in \mathbf{R}^{l \times m}$ ,  $D_i(\theta, \mu) \in \mathbf{R}^{l \times (n-1)m}$  and  $b_i(\theta, \mu) \in \mathbf{R}^l$ ; and  $u_i(x_i; x_{-i}, \theta, \mu)$  is a convex quadratic function in both  $x_i$  and  $x_{-i}$ .

Without loss of generality, each agent controls  $m$  variables, and, given the other agents’ choices and parameters  $\mu$ ,  $\theta$ , it optimizes a convex quadratic function subject to polyhedral constraints. We say that  $x^*$  is a (generalized) Nash equilibrium of (1) if it satisfies Definition III.2.

**Definition III.2** (Nash equilibrium). *A strategy profile  $x^* = (x_1^*, \dots, x_n^*)$  is a Nash equilibrium for some given  $\theta$  and  $\mu$  if, for any agent  $i$ ,  $u_i(x_i^*; x_{-i}^*, \theta, \mu) \leq u_i(x_i; x_{-i}^*, \theta, \mu)$  for any  $x_i \in \{B_i(\theta, \mu)x_i + D_i(\theta, \mu)x_{-i}^* \leq b_i(\theta, \mu)\}$ .*

We assume that the game is potential as in Definition III.3.

**Definition III.3** (Potential game [2]). *A game of the form III.1 is a potential game if there exists a function  $\Phi(x; \theta, \mu)$  such that  $x^*$  is a Nash Equilibrium of the game if and only if  $x^*$  is the optimal solution of*

$$\begin{aligned} & \underset{x \in \mathbf{R}^{mn}}{\text{minimize}} && \Phi(x; \theta, \mu) \\ & \text{subject to} && B_i(\theta, \mu)x_i + D_i(\theta, \mu)x_{-i} \leq b_i(\theta, \mu) \\ & && i = 1, \dots, n. \end{aligned} \quad (2)$$

We assume that the potential function  $\Phi(x; \theta, \mu)$  is a convex quadratic function in  $x$ ; this implies, in most cases, that the agents have quadratic utility functions. Several games in economics [27], navigation [28], driving [3] and robotics [29] involve agents with quadratic utility functions.

Let  $R$  and  $c$  be a matrix and a vector of appropriate dimensions, respectively. Then,

$$\Phi(x; \theta, \mu) = (1/2)x^T R(\theta, \mu)x + c(\theta, \mu)^T x.$$

Let the complementarity operator  $\clubsuit \perp \spadesuit$  be equivalent to  $\clubsuit^T \spadesuit = 0$ . By taking the Karush–Kuhn–Tucker conditions of the problem of Definition III.3, a vector  $x^*$  is a Nash equilibrium if and only if there exists a  $\lambda^*$  such that

$$\begin{aligned} 0 &= R(\theta, \mu)x^* + c(\theta, \mu) + A(\theta, \mu)^T \lambda^*, \\ 0 &\leq b(\theta, \mu) - A(\theta, \mu)x^* \perp \lambda^* \geq 0, \end{aligned}$$

where we define  $A(\theta, \mu)$  to be such that the inequality  $0 \leq b(\theta, \mu) - A(\theta, \mu)x$  is equivalent to  $B_i(\theta, \mu)x_i + D_i(\theta, \mu)x_{-i} \leq b_i(\theta, \mu)$  for every agent  $i = 1, \dots, n$ .

## IV. OUR METHOD

In this section, we describe the fundamental ingredients for our active-set based learning algorithm. By *active-set-based*, we mean that the algorithm identifies a subset of the active constraints and fixes them to 0 (*i.e.*, their bound) at each iteration to perform implicit differentiation of the loss function. Given a dataset  $\mathcal{D} = \{(\bar{x}^k, \bar{\mu}^k)\}_{k=1}^K$  where the agents play Nash equilibria  $\bar{x}^k$  for every  $k = 1, \dots, K$ , our algorithm learns the parameters  $\theta$  such that each  $(\bar{x}^k, \bar{\mu}^k)$  is as close to a Nash equilibrium as possible, *e.g.*, so that the data  $\mathcal{D}$  minimizes the prediction-based loss function with respect to the Nash equilibria induced by  $\theta$ . For each datapoint  $k \in \{1, \dots, K\}$ , let  $x^k = x^k(\theta, \bar{\mu}^k)$  be the Nash equilibrium induced by some parameters  $\theta \in \Theta$ , where  $\Theta$  is the space of the rationality parameters. We will consider the loss function over  $\mathcal{D}$  given by,

$$\mathcal{L}(\theta; \mathcal{D}) = (1/|\mathcal{D}|) \sum_{\bar{x}^k \in \mathcal{D}} \|x^k - \bar{x}^k\|_2^2. \quad (3)$$

To compute the points  $x^k$ , we devise a bilevel problem enforcing that  $x^k$  is a Nash equilibrium given  $\theta$ .

### A. Bilevel formulation

For the given data  $\mathcal{D}$  and a closed, convex and known set of possible parameters  $\Theta$ , we aim to find the parameters  $\theta \in \Theta$  that minimize, for each  $k = 1, \dots, K$ , the squared  $L_2$ -norm of  $x^k - \bar{x}^k$ . Equivalently, we aim to find the  $\theta$  that solves the optimization problem

$$\underset{x^k, \lambda^k, \theta}{\text{minimize}} \quad (1/K) \sum_{k=1}^K \|x^k - \bar{x}^k\|_2^2 \quad (4a)$$

$$\begin{aligned} & \text{subject to} && 0 = R(\theta, \bar{\mu}^k)x^k + c(\theta, \bar{\mu}^k) + A(\theta, \bar{\mu}^k)^T \lambda^k, \\ & && 0 \leq b(\theta, \bar{\mu}^k) - A(\theta, \bar{\mu}^k)x^k \perp \lambda^k \geq 0 \end{aligned}$$

$$x^k \in \mathbf{R}^{mn}, \lambda^k \in \mathbf{R}_+^{ln} \quad k = 1, \dots, K, \quad (4b)$$

$$\theta \in \Theta. \quad (4c)$$

The variables  $x^k \in \mathbf{R}^{nm}$  and  $\lambda^k \in \mathbf{R}_+^{ln}$  are the strategy profile and the dual variables associated with the agents' optimization problem, respectively, while  $\theta$  are the variables associated with the rationality parameters. In (4b), we enforce that the strategy profiles  $x$  are Nash equilibria of the game induced by  $\theta$  and the game parameters  $\bar{\mu}$ . In (4a), we minimize the average over data points of the squared  $L_2$ -distance between each data point  $\bar{x}^k$  and  $x^k$ . The optimization problem (4) is nonconvex due to the complementarity constraints. We can obtain an equivalent formulation by explicitly setting either the LHS or the RHS of the complementarity constraints (4b) to 0. For each data point  $k$ , we define an *active-set*  $Z^k \subseteq \{1, \dots, lm\}$ , i.e., the set of indices of tight complementarity constraints; whenever  $z \in Z^k$ , the  $z$ -th constraint  $b(\theta, \bar{\mu}^k)_z - A(\theta, \bar{\mu}^k)_z x^k = 0$  (is tight). Otherwise, if  $z \notin Z^k$ , then  $\lambda_z^k = 0$ . For a given active-set  $Z^k$ , let  $Y^k$  be its complement. Then, problem (4) is equivalent to the following minimization problem over  $Z^k, x^k, \lambda^k, \theta$ ,

$$\begin{aligned} \text{minimize} \quad & (1/K) \sum_{k=1}^K \|x^k - \bar{x}^k\|_2^2 & (5a) \\ \text{subject to} \quad & 0 = R(\theta, \bar{\mu}^k)x^k + c(\theta, \bar{\mu}^k) + A(\theta, \bar{\mu}^k)^T \lambda^k, & (5b) \\ & 0 \leq b(\theta, \bar{\mu}^k) - A(\theta, \bar{\mu}^k)x^k, \quad \lambda^k \geq 0, & (5c) \\ & 0 = b(\theta, \bar{\mu}^k)_{Z^k} - A(\theta, \bar{\mu}^k)_{Z^k} x^k, \quad 0 = \lambda_{Y^k}^k, & (5d) \\ & x^k \in \mathbf{R}^{nm}, \lambda^k \in \mathbf{R}_+^{ln}, \quad k = 1, \dots, K, \\ & \theta \in \Theta, \end{aligned}$$

where we use the subscript  $Z^k$  to refer to the set of constraints with indices contained in  $Z^k \subseteq \{1, \dots, lm\}$ , and  $Y^k = \{1, \dots, lm\} \setminus Z^k$ . In (5c), we relax the complementarity constraints of (4b) by enforcing non-negativity on both the LHS and RHS involved in the complementarity constraints; in (5d), we enforce the tightness conditions on the active-sets  $Z^k$  for any  $k$ . Although (5) does not include any complementarity constraints, it involves exponentially many choices of  $Z^k$  for any  $k$ . However, given some fixed  $Z = (Z^1, \dots, Z^K)$ , we can employ (5) to formulate the loss function (3) and devise a differentiation method to iteratively refine the estimates of  $\theta$ .

### B. Implicit differentiation of the loss function

We exploit the gradient information of the loss function (3) associated with problem (5). Let  $x^k = x_{Z^k}^k(\theta, \bar{\mu}^k)$  (we omit  $Z^k$  when the context is clear) be implicitly defined as

$$F(x^k, \lambda^k; \theta) = \begin{bmatrix} R(\theta, \bar{\mu}^k)x^k + c(\theta, \bar{\mu}^k) + A(\theta, \bar{\mu}^k)^T \lambda^k \\ b(\theta, \bar{\mu}^k)_{Z^k} - A_{Z^k} x^k \\ \lambda_{Y^k} \end{bmatrix} = 0, \quad (6)$$

i.e.,  $x^k$  solves  $0 = F(x^k, \lambda^k; \theta)$ , or, in other words,  $x^k$  is a Nash equilibrium given  $Z$ . Therefore, for a given active-set  $Z$ , we reformulate the loss function (5) over  $\mathcal{D}$  as

$$\mathcal{L}(\theta, Z; \mathcal{D}) = (1/|\mathcal{D}|) \sum_{\bar{x}^k \in \mathcal{D}} \|x_{Z^k}^k - \bar{x}^k\|_2^2. \quad (7)$$

We evaluate the gradient of  $\mathcal{L}(\theta, Z; \mathcal{D})$  with respect to  $\theta$  as

$$\nabla_{\theta} \mathcal{L}(\theta, Z; \mathcal{D}) = (1/|\mathcal{D}|) \sum_{\bar{x}^k \in \mathcal{D}} 2 \nabla_{\theta} (x_{Z^k}^k)^T (x_{Z^k}^k - \bar{x}^k). \quad (8)$$

Equivalently, we can compute (8) by employing the implicit function theorem [30, Theorem 1B.1], as in Theorem IV.1.

**Theorem IV.1** (Implicit function theorem). *Let  $G(\beta)$  be implicitly defined by the relation  $0 = H(\alpha, \beta)$  for  $H: \mathbf{R}^{d_1} \times \mathbf{R}^{d_2} \mapsto \mathbf{R}^{d_1}$ . Assume 1)  $H$  is continuously differentiable*

*in a neighborhood of  $(\alpha_0, \beta_0)$ , 2)  $H(\alpha_0, \beta_0) = 0$ , and 3)  $\nabla_{\alpha} H(\alpha_0, \beta_0)$  is nonsingular. Then,  $G$  is continuously differentiable in a neighborhood of  $(\alpha_0, \beta_0)$  with Jacobian,*

$$\nabla_{\beta} G(\beta) = \nabla_{\alpha} H(\alpha_0, \beta_0)^{-1} \nabla_{\beta} H(\alpha_0, \beta_0).$$

We want to differentiate  $x^k$  and  $\lambda^k$  with respect to  $\theta$ , where  $x^k$  and  $\lambda^k$  are defined implicitly by the relation (6). We, therefore, let  $\alpha = (x^k, \lambda^k)$ ,  $\beta = \theta$  and  $H(\alpha, \beta) = H((x^k, \lambda^k), \theta) = F(x^k, \lambda^k; \theta)$  so that  $H(\alpha, \beta)$  is equal to

$$\begin{bmatrix} R(\theta, \bar{\mu}^k) & A(\theta, \bar{\mu}^k)^T \\ -A(\theta, \bar{\mu}^k)_{Z^k} & 0 \\ 0 & I_{Y^k} \end{bmatrix} \begin{bmatrix} x^k \\ \lambda^k \end{bmatrix} - \begin{bmatrix} c(\theta, \bar{\mu}^k) \\ b_{Z^k}(\theta, \bar{\mu}^k) \\ 0 \end{bmatrix}. \quad (9)$$

Using (9), Theorem IV.1 allows us to differentiate  $(x^k, \lambda^k)$  with respect to  $\theta$ , and, therefore, to differentiate the loss function (7) with respect to  $\theta$  for any active-set  $Z$  of (5).

### C. The algorithm

We employ the ingredients of Sections IV-A and IV-B to create a stochastic first-order method to learn the parameters  $\theta$  given the dataset  $\mathcal{D}$ . We formalize our approach in Algorithm 1. Given an instance of a game of Definition III.1, the dataset  $\mathcal{D}$ , a maximum number of iterations  $T$ , and a series of learning rates  $\{\eta_t\}_{t=1}^T$ , the algorithm returns the estimated rationality parameters  $\theta_T$ .

---

#### Algorithm 1 Active-set algorithm

---

**Require:** The data  $\mathcal{D} = \{(\bar{x}^k, \bar{\mu}^k)\}_{k=1}^K$ , the maximum number of iterations  $T$ , and step sizes  $\{\eta_t\}_{t=1}^T$

- 1: Initialize  $\theta_0$
- 2: **for**  $t = 0, \dots, T-1$  **do**
- 3: Choose  $(\bar{x}^k, \bar{\mu}^k)$  uniformly from  $\mathcal{D}$
- 4:  $x^{(t)}, \lambda^{(t)} \leftarrow$  primal and dual variables from (2)
- 5:  $Z^{(t)} \leftarrow \{z : b(\theta^{(t)}, \bar{\mu}^k)_z - A(\theta^{(t)}, \bar{\mu}^k)_z x^{(t)} = 0\}$
- 6:  $Y^{(t)} \leftarrow \{z : \lambda_z^{(t)} = 0\}$
- 7: Set  $\mathcal{L}_t(x) \leftarrow \mathcal{L}(\theta^{(t)}, Z^{(t)}; \{\bar{x}^k\})$
- 8:  $\nabla_{\theta} \mathcal{L}_t(x^{(t)}) = \nabla_{\theta} (x^{(t)}(\theta^{(t)}))^T \nabla_x \mathcal{L}_t(x^{(t)}(\theta^{(t)}))$
- 9:  $\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta_t \nabla_{\theta} \mathcal{L}_t(x^{(t)})$
- 10: **end for**
- 11: **return**  $\theta^{(T)}$

---

After initializing  $\theta^{(0)}$  (line 1), at each step  $t$ , the algorithm picks a random datapoint  $(\bar{x}^k, \bar{\mu}^k)$  from  $\mathcal{D}$  with uniform probability (line 3). Given  $(\bar{x}^k, \bar{\mu}^k)$ , the algorithm solves the optimization problem in line 2 to compute  $x^{(t)}$  and  $\lambda^{(t)}$ , with  $x^{(t)}$  being a Nash equilibrium of the game induced by  $\theta^{(t)}$  and  $\bar{\mu}^k$  (line 4). Let  $Z^{(t)} = \{z : b(\theta^{(t)}, \bar{\mu}^k)_z - A(\theta^{(t)}, \bar{\mu}^k)_z x^{(t)} = 0\}$  be the active-set at  $x^{(t)}$ , and  $Y^{(t)} = \{z : \lambda_z^{(t)} = 0\}$  (lines 5 and 6). On the one hand, if  $Z^{(t)} \cap Y^{(t)} = \emptyset$ , i.e., there is no degeneracy in the complementarities, then the matrix in Step 9 is invertible. We can obtain the gradient of  $x^{(t)}$  with respect to  $\theta^{(t)}$  using Theorem IV.1 (line 8). We use this gradient information to update the estimates for  $\theta^{(t)}$  to reduce the loss proportionally to  $\eta_t$ , and we continue to iterate (line 9). On the other hand, if  $Z^{(t)} \cap Y^{(t)} \neq \emptyset$ , we have a degenerate solution. In the following subsection, we explain how to retrieve gradient information in degenerate active-sets. When the algorithm reaches its iteration limit, it returns the last estimate  $\theta^{(T)}$ .

#### D. Degenerate active-set

Whenever, at some step  $t$ ,  $Z^{(t)} \cap Y^{(t)} \neq \emptyset$ , we cannot obtain the gradient from (9) since the matrix is not invertible. This implies we must choose an alternative non-degenerate active-set on which we evaluate the gradient (8). We do so using the following rule, which is motivated by the theory in Section V.

**Rule 1.** Suppose we are at iteration  $t$  of Algorithm 1. Update  $\theta^{(t+1)}$  using the gradient  $\nabla_{\theta} \mathcal{L}(\tilde{\theta}; \{\bar{x}^k\})$ , where  $\tilde{\theta}$  is chosen uniformly on  $B_{\epsilon}(\theta^{(t)})$ , i.e., the Euclidean ball of radius  $\epsilon$ . If it is the case that at  $\tilde{\theta}$ ,  $Z^{(t)} \cap Y^{(t)} \neq \emptyset$ , let  $W^{(t)} = Z^{(t)} \cap Y^{(t)}$ . Randomly and uniformly partition  $W^{(t)}$  into two sets,  $W_1^{(t)}$  and  $W_2^{(t)}$ . Let  $\tilde{Z}^{(t)} = Z^{(t)} \setminus W_1^{(t)}$  and  $\tilde{Y}^{(t)} = Y^{(t)} \setminus W_2^{(t)}$ . Use  $\mathcal{L}_t(x) \leftarrow \mathcal{L}(\tilde{\theta}, \tilde{Z}^{(t)}; \{x_k\})$  in Step 7 of Algorithm 1.

Essentially, Rule 1 updates the gradient at  $\theta^{(t)}$  using the gradient information from a  $\tilde{\theta}$ , a slightly perturbed version of  $\theta^{(t)}$ . In the presence of degeneracy, Rule 1 randomly partitions the intersection (i.e. the elements causing the degeneracy) of  $Z$  and  $Y$  and uses this active-set to update the gradient in Step 8. However, we note that due to the perturbation, degeneracy rarely occurs in practice.

#### V. CONVERGENCE ANALYSIS

In this section, we will provide a proof of the convergence behavior of Algorithm 1. In particular, we will prove that a smoothed version of the gradient of the loss with respect to  $\theta^{(t)}$  at iteration  $t$  converges to zero (Theorem V.1), and, therefore, that our algorithm eventually finds either a local minimum of the smoothed loss or a saddle point. While this does not guarantee convergence to a global minimum, since we are providing an algorithm designed to run on a large dataset, we argue that a local minimum can empirically provide a good generalization of  $\theta$ , as we will show in our computational experiments. For notational simplicity, we will let  $A(\theta, \mu)$  be  $A$ . First, for any  $\theta$  and any non-degenerate active-set  $Z$  (i.e.,  $Z \cap Y = \emptyset$ ), (7) provides an explicit expression for  $\nabla_{\theta} \mathcal{L}(\theta; Z, \{\bar{x}_k\})$  at any point  $\bar{x}_k \in \mathcal{D}$ . Specifically,  $\nabla_{\theta} \mathcal{L}(\theta; Z, \mathcal{D})$  is given by,

$$\nabla_{\theta} \mathcal{L}(\theta; Z, \mathcal{D}) = \frac{2}{|\mathcal{D}|} \sum_{k=1}^K S_k(Z_k, \theta)^{-T} \begin{bmatrix} x_{Z_k}^k - \bar{x}^k \\ 0 \end{bmatrix},$$

$$S_k(Z_k, \theta) = \begin{bmatrix} R(\theta, \bar{\mu}^k) & A^T \\ -A_{Z_k} & 0 \\ 0 & I_{Y_k} \end{bmatrix},$$

and the  $j$ -th row of the Hessian  $\nabla^2 \mathcal{L}(\theta, Z, \mathcal{D})_j$  is given by,

$$\frac{2}{|\mathcal{D}|} \sum_{k=1}^K S_k(Z_k, \theta)^{-T} \partial_{\theta_j} S_k(Z_k, \theta) S_k(Z_k, \theta)^{-T} \begin{bmatrix} x_{Z_k}^k - \bar{x}^k \\ 0 \end{bmatrix}.$$

To prove our result, we will assume the following.

**Assumption 1** (Constraint qualifications).

- 1) The linear independence constraint qualifications (LICQs) hold for the set  $\{x \mid b(\theta, \bar{\mu}^k) - Ax \geq 0\}$  for every  $\theta \in \Theta$ , and  $k = 1 \dots K$ ,
- 2) the set  $\{x \mid Ax \leq b(\theta, \mu)\}$  is uniformly bounded in  $x$  for all  $\theta, \mu$ , and
- 3) for almost all  $\theta \in \Theta$  (in the Lebesgue-measure sense),  $Z_k(\theta) \cap Y_k(\theta) = \emptyset$  for every  $k = 1 \dots K$ .

**Assumption 2** (Well-behaved derivatives).

- 1) there exists an  $L_0 > 0$  such that,  $\|S_k(Z_k, \theta)^{-T}\|_{\infty} < L_0$  for every  $\theta \in \Theta$ ,  $Z_k \subseteq \{1, 2, \dots, lm\}$  and  $k = 1, \dots, K$ , and

- 2) the function  $\partial_{\theta_j} R(\theta, \bar{\mu}^k)$  is bounded, namely, there exists an  $L_3 > 0$  such that for all  $k = 1, \dots, K$  and all  $\theta \in \Theta$ ,  $\|\partial_{\theta_j} R(\theta, \bar{\mu}^k)\|_{\infty} \leq L_3$ .

Assumptions 1 and 2 are satisfied in several practical applications; for instance, in Proposition V.1, we prove these assumptions hold for the games we consider in Section VI.

**Proposition V.1.** Suppose that  $b(\theta, \mu) = b(\mu)$  has no dependency on  $\theta$ ,  $A$  is constant, and  $R$  and  $c$  are affine functions of  $\theta$ , i.e.,

$$R(\theta, \mu) = R_0(\mu) + \sum_{i=1}^p R_i(\mu) \theta_i, \quad c(\theta, \mu) = C(\mu) \theta,$$

with  $R_0(\mu) \succ 0$  and  $R_i(\mu) \succeq 0$  for  $i = 1, \dots, p$ . If Assumption 1 holds, then Assumption 2 holds.

*Proof.* First, we prove that, under our assumptions,  $S_k(Z_k, 0)$  is an invertible matrix. Otherwise, there would exist an  $(x_0, \lambda_0)$  such that  $S_k(Z_k, 0)(x_0, \lambda_0)^T = 0$  and therefore  $x_0 = -R(0, \bar{\mu}^k)^{-1} A^T \lambda_0$ , meaning that,

$$\begin{bmatrix} A_{Z_k} R(0, \bar{\mu}^k)^{-1} A^T \\ I_{Y_k} \end{bmatrix} \lambda_0 = 0.$$

This is equivalent to,

$$\lambda_{0_{Y_k}} = 0, \quad A_{Z_k} R(0, \bar{\mu}^k)^{-1} A_{Z_k}^T \lambda_{0_{Z_k}} = 0,$$

and since, by Assumption 1, the matrix  $A_{Z_k} R(0, \bar{\mu}^k)^{-1} A_{Z_k}^T$  is invertible. This implies that  $\lambda_0 = 0$  and  $x_0 = 0$ , which is a contradiction. Now we will prove that the infinity norm of  $S_k(Z_k, \theta)$  is bounded. Let  $M_0$  and  $M_1(\theta)$  be defined as

$$M_0 = \begin{bmatrix} R_0(\mu) & A^T \\ -A_{Z_k} & 0 \\ 0 & I_Y \end{bmatrix}, \quad M_1(\theta) = \begin{bmatrix} \sum_{i=1}^p R_i(\mu) \theta_i & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

so that  $S_k(Z_k, \theta) = M_0 + M_1(\theta)$ . Then

$$S_k(Z_k, \theta)^T S_k(Z_k, \theta) = M_0^T M_0 + \sum_{i=1}^p \theta_i^2 R_i^T R_i + \sum_{i=1}^p \theta_i (R_i^T R_0 + R_0^T R_i). \quad (10)$$

Since  $R_i$  is positive semi-definite,  $R_0$  is positive-definite, and  $M_0$  is an invertible matrix, (10) is positive-definite. Furthermore, its smallest eigenvalue is at least as large as the minimum eigenvalue  $\epsilon_0$  of  $M_0^T M_0$ . Therefore, the largest eigenvalue of  $(S_k(Z_k, \theta)^T S_k(Z_k, \theta))^{-1}$  is  $1/\epsilon_0$  and so the greatest singular value of  $S_k(Z_k, \theta)^{-T}$  is  $1/\sqrt{\epsilon_0}$ . This means that the infinity norm of  $S_k$  must be bounded.  $\square$

We remark that  $\nabla_{\theta} \mathcal{L}(\tilde{\theta}; \{\bar{x}^k\})$  is well-defined by Assumption 2 because  $\tilde{\theta}$  is a continuous random variable and the gradient is defined almost-everywhere. For any fixed  $\epsilon$  and  $\theta_0$ , let  $g$  be,

$$g(\theta) = (1/K) \sum_{k=1}^K \int_{\theta_0}^{\theta} \int_{B_{\epsilon}(v)} \frac{1}{\pi \epsilon^2} \nabla \mathcal{L}(u; \{\bar{x}^k\}) du dv,$$

where the outer integral can be taken along any path from  $\theta_0$  to  $\theta$  since  $\nabla \mathcal{L}(u; \{\bar{x}^k\})$  is potential.  $g$  can then be interpreted as a smoothed version of  $\mathcal{L}$ . We will show that Algorithm 1 with the update rule 1 is equivalent to performing gradient descent on the smooth, nonconvex function  $g$ ; this will allow us to provide our main convergence result.

**Theorem V.1.** For appropriately given step-sizes  $\{\eta_t\}_{t=1}^T$  given by  $\eta_t = 1/t^{(3/4)}$ , the active-set algorithm with gradient Rule 1 satisfies  $\lim_{T \rightarrow \infty} \mathbf{E}[\|\nabla g(\theta_T)\|_2] = 0$ .

*Proof.* We will begin by bounding relevant derivatives of  $g$ . Since terms in the relevant expressions for each derivative are bounded (using the boundedness of  $S_k(S_k, \theta)$  and

$\partial_\theta R(\theta, \bar{\mu}^k)$ , there exist some constants  $L_1$  and  $L_2 > 0$  such that, for all  $k = 1 \dots K$  and  $Z \subseteq \{1, \dots, lm\}$ ,

$$\|\nabla \mathcal{L}(\theta; Z, \{X_k\})\|_2 \leq L_1, \quad -L_2 \preceq \nabla^2 \mathcal{L}(\theta, Z, \{X_k\}) \preceq L_2.$$

Note that  $\mathbf{E}[\nabla \mathcal{L}(\tilde{\theta})] = \nabla g(\theta)$ , where the randomness stems from the choice of the datapoint and the update Rule 1. By the claim,  $g$  is Lipschitz continuous with constant  $L_1$  since,

$$\begin{aligned} \|\nabla g(\theta)\|_2 &= \left\| \left( \frac{1}{K} \sum_{k=1}^K \frac{1}{\pi \epsilon^2} \int_{B_\epsilon(\theta)} \nabla \mathcal{L}(u; \{X_k\}) du \right) \right\|_2 \\ &\leq \left( \frac{1}{K} \sum_{k=1}^K \frac{1}{\pi \epsilon^2} \int_{B_\epsilon(\theta)} \|\nabla \mathcal{L}(u; \{X_k\})\|_2 du \right) \\ &\leq L_1. \end{aligned}$$

This and the Cauchy-Schwarz inequality also imply that,

$$\mathbf{E}[\|\nabla \tilde{\mathcal{L}}(\theta) - \nabla g(\theta)\|_2^2] \leq 4L_1^2.$$

Since  $\mathcal{L}$  is twice-differentiable on any  $Z$ , we have that

$$\nabla^2 g(\theta) = \left( \frac{1}{K} \sum_{k=1}^K \frac{1}{(\pi \epsilon^2)} \int_{B_\epsilon(\theta)} \nabla^2 \mathcal{L}(u; \{X_k\}) du \right).$$

Since  $\nabla^2 g$  is an average of  $\nabla^2 \mathcal{L}$ , we have that  $-L_2 \preceq \nabla^2 g(\theta) \preceq L_2$ . Using the facts above, the active-set algorithm with gradient Rule 1 mirrors a stochastic gradient descent on  $g$ , a Lipschitz-continuous function with bounded second derivatives. We can then use a standard theorem about the convergence of stochastic gradient descent on smooth nonconvex functions such as [31, proposition 3].  $\square$

Using the definition of  $g$  again, we have shown that,

$$\lim_{T \rightarrow \infty} \mathbf{E}_{\theta(T)} [\|\mathbf{E}_{\mathcal{U}}[\nabla_\theta \mathcal{L}(\theta(T) + \mathcal{U}; \mathcal{D})]\|_2] = 0,$$

where the outer expectation is taken over the steps of the algorithm, and the inner expectation is taken over the uniform distribution on the Euclidean ball  $\mathcal{U} \sim U[B_\epsilon(0)]$ .

## VI. COMPUTATIONAL RESULTS

In this section, we showcase the effectiveness of our algorithm via some computational experiments. We run the experiments on a *Intel Xeon Gold 6246R* with 16GB of RAM, and we use *Gurobi 9.5* [32] as the optimization solver. For each experiment, we generate a training dataset  $\mathcal{D}_{\text{train}}$  with 90 samples, and a test dataset for evaluation  $\mathcal{D}_{\text{test}}$  with 10 datapoints. In both the datasets, the datapoints  $\bar{x}_k$  are Nash equilibria given  $\bar{\theta}, \bar{\mu}_k$  with the addition of an independent Gaussian error vector  $\epsilon_k \sim \mathcal{N}(0, \sigma)$  with  $\sigma = 0.001$ . We use a smoothing parameter  $\epsilon = 0.01$  for Rule 1. We remark that the true rationality parameters  $\theta$  are unknown. To assess the performance of Algorithm 1 given the estimated parameters  $\theta^*$ , we employ the test error

$$\sqrt{\frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(\bar{\mu}_k, \bar{x}_k) \in \mathcal{D}_{\text{test}}} \|x(\theta^*; \bar{\mu}_k) - \bar{x}_k\|_2^2}.$$

### A. Cournot Games

A generalized Cournot game is a simultaneous non-cooperative game where a series of agents decide the quantity of a homogeneous product to inject into a market. For every agent  $i$ , let the variable  $x_i \geq 0$  represent the quantity of product produced, and the parameter  $c_i \in \mathbf{R}$  be the unit-cost of production. The price of the product is determined by the *inverse demand function*  $F(x) = a - b \sum_{j=1}^n x_j$ , where  $\theta = (a, b)$  and  $a, b \in \mathbf{R}$ . The utility (cost) function of  $i$  is

$$u_i(x_i; x_{-i}, \theta, \mu) = -F(x)x_i + c_i x_i,$$

*i.e.*, the revenues due to selling  $x_i$  units at price  $F$  minus the production cost  $c_i x_i$ . We define  $\mu = (c_1, \dots, c_n)$  and the potential function  $\Phi(x; \theta, \mu)$  as

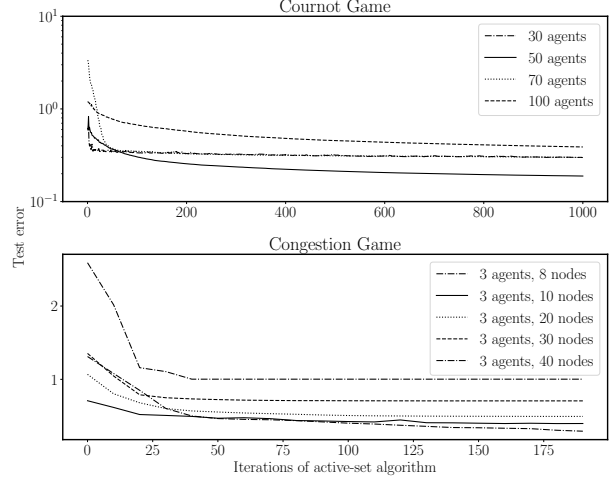


Fig. 1. Test error for the Cournot and congestion games.

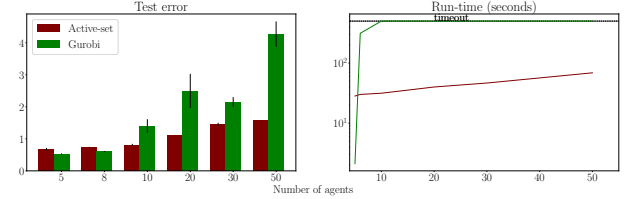


Fig. 2. Comparison to Gurobi for the Cournot game.

$$c^T x + bx^2 + b \sum_{1 \leq i < j \leq n} x_i x_j - a^T x.$$

We consider  $n = 30, 50, 70$ , and 100 agents, and we randomly generate  $\bar{\theta}_k$  and  $\bar{\mu}_k$  according to the modulus of a Gaussian distribution  $\mathcal{N}(0, \sigma)$  with  $\sigma = 1$ . We report the test error for the Cournot instances in the top half of Figure 1. In all three cases, Algorithm 1 learns accurate rationality parameters  $\theta$  inducing low test errors. Furthermore, in Figure 2, we compare the running times and the test errors of Algorithm 1 with Gurobi solving the nonconvex problem (4). Since the solver runs out of time on our smallest instance with  $n = 50$ , we generated some smaller instances with  $n \in \{5, 8, 10, 20, 30, 50\}$ . We also reduced the size of the dataset, making it equal to the number of players, because Gurobi could not solve the problem with 100 data points in a reasonable time for any number of players. While Gurobi times out (time limit of 2000 seconds) whenever  $n \geq 10$ , our algorithm solves all the instances.

### B. Congestion games

We consider a class of congestion games [28] defined over a direct graph  $G = (V, E)$ , where  $V$  is the set of vertices and  $E$  is the set of edges. Each agent  $i$  has to route  $d_i \in \mathbf{R}$  units of traffic from a node  $s_i \in V$  to a destination  $t_i \in V$ . Let the variable  $x_{ie}$  represent the amount of traffic agent  $i$  routes through edge  $e \in E$ , and  $C_e \in \mathbf{R}$  be the cost parameter for edge  $e$ . Let  $\bar{C} = (C_{e_1}, \dots, C_{e_{|E|}})$  be the vector of edge costs and  $L \in \mathbf{R}^{|E| \times p}$  be a matrix of factors relating to each edge (*e.g.*, the length, width, and elevation of a road or the frequency of public transport). We assume that  $C = L\theta$  for some unknown rationality parameters  $\theta \in \mathbf{R}^p$  and  $\mu = (s_i, t_i, d_i)_{i \in I}$ . All considered, each agent  $i$  minimizes

$$u_i(x_i; x_{-i}, \theta, \mu) = \sum_{e \in E} C_e x_{ie} (x_{1e} + \dots + x_{ne}),$$

subject to the standard flow constraints on  $G$ . The potential function of the game is

$$\Phi(x; \theta, \mu) = \sum_{e \in E} C_e \sum_{i=1}^n ((1/2)x_{ie}^2 + \sum_{j \neq i} x_{ie} x_{je}).$$

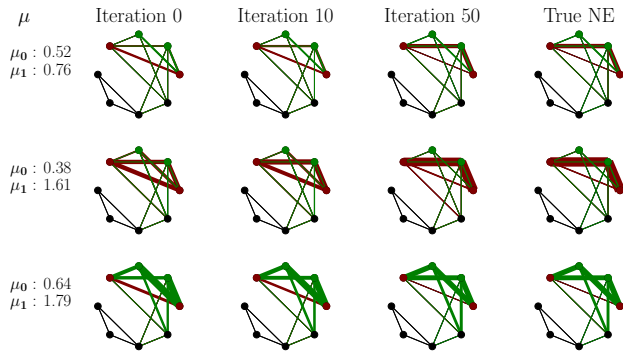


Fig. 3. Three instances with different  $\mu$  of a small congestion game with  $n = 2$  among the red and the green agent. The edge thickness represents the amount of traffic routed through the edge. *True NE* represents the ground-truth Nash equilibrium.

We generate instances with  $n = 3$  and  $|V| \in \{8, 10, 20, 30, 40\}$ . The graphs  $G$  are Erdős–Rényi random graphs where each edge has a probability  $p = 0.3$  of being present. Finally, we randomly generate  $\theta_k$  according to the modulus of a Gaussian distribution  $\mathcal{N}(0, \sigma)$  with  $\sigma = 1$ . In Figure 1, we report the test errors on the congestion games. Similarly to the Cournot games, our algorithm converges to small test errors. In Figure 3, we provide a small toy example on a congestion game with  $n = 2$ .

## VII. CONCLUSIONS

We introduced a stochastic first-order algorithm to estimate the parameters of potential games. Our active-set-based algorithm alternates between an active-set step enforcing that the parameters induce Nash equilibria and an implicit-differentiation step to update the estimates of the parameters. From a theoretical perspective, we proved that our algorithm is guaranteed to converge. From a practical perspective, we demonstrated the efficiency of our technique via computational tests on Cournot and congestion games.

## REFERENCES

- [1] J. F. Nash, “Equilibrium Points in  $n$ -Person Games,” *Proc. of the National Acad. of Sciences of the United States of America*, vol. 36, no. 1, pp. 48–49, 1950.
- [2] D. Monderer and L. S. Shapley, “Potential Games,” *Games and Economic Behavior*, vol. 14, no. 1, pp. 124–143, 1996.
- [3] A. Zanardi, E. Mion, M. Bruschetta, S. Bolognani, A. Censi, and E. Frazzoli, “Urban driving games with lexicographic preferences and socially efficient nash equilibria,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4978–4985, 2021.
- [4] C. Daskalakis, M. Fishelson, and N. Golowich, “Near-optimal no-regret learning in general games,” *NIPS*, vol. 34, pp. 27 604–27 616, 2021.
- [5] J. Hartline, V. Syrgkanis, and E. Tardos, “No-regret learning in bayesian games,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS’15. Cambridge, MA, USA: MIT Press, 2015, p. 3061–3069.
- [6] R. Kleinberg, G. Piliouras, and É. Tardos, “Multiplicative updates outperform generic no-regret learning in congestion games,” in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 533–542.
- [7] W. Krichene, B. Drighès, and A. M. Bayen, “Online learning of nash equilibria in congestion games,” *SIAM J. on Control and Optimization*, vol. 53, no. 2, pp. 1056–1081, 2015.
- [8] F. Fabiani and A. Bemporad, “A learning-based approach to multi-agent decision-making,” *arXiv:2212.12561*, 2022.
- [9] J. Li, J. Yu, Y. Nie, and Z. Wang, “End-to-end learning and intervention in games,” *NIPS*, vol. 33, pp. 16 653–16 665, 2020.

- [10] H. Heaton, D. McKenzie, Q. Li, S. W. Fung, S. Osher, and W. Yin, “Learn to Predict Equilibria via Fixed Point Networks,” *arXiv:2106.00906*, 2021.
- [11] D. Bertsimas, V. Gupta, and I. C. Paschalidis, “Data-driven estimation in equilibrium using inverse optimization,” *Mathematical Programming*, vol. 153, no. 2, pp. 595–633, 2015.
- [12] S. Allen, S. A. Gabriel, and J. P. Dickerson, “Using inverse optimization to learn cost functions in generalized nash games,” *Computers & Operations Research*, vol. 142, p. 105721, 2022.
- [13] Y. Vorobeychik, M. P. Wellman, and S. Singh, “Learning payoff functions in infinite games,” *Machine Learning*, vol. 67, no. 1, pp. 145–168, 2007.
- [14] C. K. Ling, F. Fang, and J. Z. Kolter, “What Game Are We Playing? End-to-end Learning in Normal and Extensive Form Games,” in *IJCAI*, 2018, pp. 396–402.
- [15] K. Waugh, B. D. Ziebart, and J. A. Bagnell, “Computational rationalization: The inverse equilibrium problem,” in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML’11. Madison, WI, USA: Omnipress, 2011, p. 1169–1176.
- [16] F. Facchinei, A. Fischer, and V. Piccialli, “On generalized Nash games and variational inequalities,” *Operations Research Letters*, vol. 35, no. 2, pp. 159–164, 2007.
- [17] G. Dragotto, “Mathematical Programming Games,” Ph.D. dissertation, Polytechnique Montréal, 2022.
- [18] A. Y. Ng and S. J. Russell, “Algorithms for inverse reinforcement learning,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML ’00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, p. 663–670.
- [19] J. Wu, W. Shen, F. Fang, and H. Xu, “Inverse game theory for stackelberg games: the blessing of bounded rationality,” *arXiv:2210.01380*, 2022.
- [20] X. Liu, L. Peters, and J. Alonso-Mora, “Learning to Play Trajectory Games Against Opponents with Unknown Objectives,” *arXiv:2211.13779*, 2022.
- [21] J. Grover, Y. Lyu, W. Luo, C. Liu, J. Dolan, and K. Sycara, “Semantically-aware pedestrian intent prediction with barrier functions and mixed-integer quadratic programming,” *IFAC-PapersOnLine*, vol. 55, no. 41, pp. 167–174, 2022, 4th IFAC Workshop on Cyber-Physical and Human Systems CPHS 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896323001283>
- [22] A. Agrawal, S. Barratt, and S. Boyd, “Learning convex optimization models,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 8, pp. 1355–1364, 2021.
- [23] W. Jin, A. Aydinoglu, M. Halm, and M. Posa, “Learning Linear Complementarity Systems,” in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 1137–1149.
- [24] P. Geiger and C.-N. Strachle, “Learning game-theoretic models of multiagent trajectories using implicit layers,” *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6), 4950–4958, 2021.
- [25] N. Zucchet and J. Sacramento, “Beyond backpropagation: bilevel optimization through implicit differentiation and equilibrium propagation,” *Neural Computation*, vol. 34, no. 12, pp. 2309–2346, 2022.
- [26] M. Arbel and J. Mairal, “Amortized implicit differentiation for stochastic bilevel optimization,” in *ICLR*, 2022.
- [27] N. S. Lambert, G. Martini, and M. Ostrovsky, “Quadratic games,” National Bureau of Economic Research, Working Paper 24914, August 2018.
- [28] R. W. Rosenthal, “A class of games possessing pure-strategy Nash equilibria,” *International Journal of Game Theory*, vol. 2, no. 1, pp. 65–67, 1973.
- [29] D. Fridovich-Keil, E. Ratner, L. Peters, A. Dragan, and C. Tomlin, “Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games,” in *2020 IEEE international conference on robotics and automation (ICRA)*, 05 2020, pp. 1475–1481.
- [30] R. T. R. Asen L. Dontchev, *Implicit Functions and Solution Mappings*. Springer Monographs in Mathematics, 2009.
- [31] D. Bertsekas and J. Tsitsiklis, “Gradient convergence in gradient methods with errors,” *SIAM Journal on Optimization*, vol. 10, 1999.
- [32] Gurobi Optimization, LLC, “Gurobi Optimizer Reference Manual,” 2023. [Online]. Available: <https://www.gurobi.com>