2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Equilibrium Seeking in Learning-Based Noncooperative Nash Games

Paolo Scarabaggio, Nicola Mignoni, Raffaele Carli, and Mariagrazia Dotoli

Abstract— **Traditionally, based on convexity, multi-agent decision-making models can hardly handle scenarios where agents' cost functions defy this assumption, which is specifically required to ensure the existence of several equilibrium concepts. More recently, the advent of machine learning (ML), with its inherent non-convexity, has changed the conventional approach of pursuing convexity at all costs. This paper explores and integrates the robustness of game theoretic frameworks in managing conflicts among agents with the capacity of ML approaches, such as deep neural networks (DNNs), to capture complex agent behaviors. Specifically, we employ feed-forward DNNs to characterize agents' best response actions rather than modeling their goals with convex functions. We introduce a technical assumption on the weight of the DNN to establish the existence and uniqueness of Nash equilibria and present two distributed algorithms based on fixed-point iterations for their computation. Finally, we demonstrate the practical application of our framework to a noncooperative community of smart energy users under a dynamic time-of-use energy pricing scheme.**

I. Introduction

Convexity (or concavity) is the cornerstone of equilibrium theory. From the first works by Von Neumann to the fundamental contributions by Nash [1], mathematical elegance and theoretical tractability have steered multi-agent decision-making models in the safe harbor of convexity. Indeed, modern game theory began with Von Neumann's result [2] establishing equilibrium existence in two-player zero-sum games, under the assumption that each players payoff is concave with respect to their strategy – equivalently, that their cost is convex with respect to their strategy.

Under this assumption, computing equilibria is, in fact, equivalent to convex programming, making the powerful tools of convex optimization readily applicable [3]. Actually, thanks to convexity, the convergence of several classes of multi-agent dynamics (centralized, decentralized, and distributed) to equilibrium can be guaranteed [4]–[8]. While (quasi) convexity is crucial in facilitating the existence of various equilibria, this assumption proves to be overly restrictive. Likewise, noncooperative games involving non-convex cost functions have garnered some attention in recent works [9], [10]. Among the works, let us mention the equilibrium notions of weak Nash Equilibrium (NE) [11], local NE (LNE) [12], generalized equilibrium [13], and critical NE [14]. However, progress in these areas has been restrained, not due to a lack of interest in non-convex settings, but rather because proving the existence and convergence to equilibria is challenging, thus diminishing their value.

The search for convexity at all costs has recently gone against the advent of machine learning (ML) with its inherent and pervasive non-convexity [15]. Indeed, ML's attitude of embracing non-convexity has led to groundbreaking advances in significant challenges, such as speech and image recognition, text generation, and many more [16], [17].

In recent years, ML has been rapidly expanding its scope to the domain of game theory, emphasizing the importance of studying non-convex games. Many outstanding challenges in this field, such as training deep neural networks (DNN) that are robust to adversarial attacks, training Generative Adversarial Networks (GANs) [18], and Multi-Agent Reinforcement Learning (MARL) [19] have been defined as multi-player games with utility functions that are non-convex in agents strategies. Among these, DNNs have proven to be successful in many prediction tasks, particularly in predicting decisions [20]. Thus, they are the perfect candidates for approximating the behavior of agents in a strategic interaction framework such as a game. Nevertheless, despite their poor representation capability, agents are still identified and controlled in most papers based on simple linear models.

The increasing complexity of agents' behavior and the practical necessity of representing them with convex cost functions are the starting points of our work. We aim at harmonizing the effectiveness of game theoretic frameworks in managing conflicting scenarios, where resources are shared among a group of agents, with the capacity of ML, specifically DNNs, to approximate the complex behaviors of agents. Unlike the state-of-the-art, we do not model the agents' behaviors as it is traditionally done, with cost functions representing their ultimate goals. Instead, we approximate these behaviors using a DNN to characterize their response actions. By introducing a technical assumption on the weight of the DNN, we demonstrate the existence of an equilibrium and its uniqueness, under additional assumptions. To compute these equilibria, we define two algorithms based on fixed-point iterations and demonstrate their convergence. Finally, we apply our theoretical results to a noncooperative community of smart energy users under a dynamic time-of-use energy pricing scheme.

The rest of the paper is organized as follows. In Section II we recall some preliminaries. In Section III we introduce the novel game theoretic framework with the integration of ML. Section IV discusses the existence and uniqueness of equilibria, while Section V shows two distributed algorithms for reaching such equilibria. In Section VI we show the illustrative application of our framework. Section VII concludes the work.

II. Preliminaries

A. Basic Notation

$\mathbb{R}^n$ denotes the set of real n-dimensional vectors while $\mathbb{N}$ denotes the set of natural numbers. $\mathbf{A}^\top$ denotes the transpose of matrix $\mathbf{A}$. The norm induced by matrix $\mathbf{A} \succeq 0$ is denoted with $\|\mathbf{x}\|_{\mathbf{A}}$, for any $\mathbf{x} \in \mathbb{R}^n$. The square norm is simply $\|\mathbf{x}\|$. The identity matrix and all-ones vector are denoted as $\mathbf{I}$ and $\mathbf{1}$, respectively. Positive (negative) semidefinite matrices are denoted with $\mathbf{A} \succeq 0$ ($\mathbf{A} \preceq 0$). For a generic set $\mathcal{X}$, its cardinality is defined by $|\mathcal{X}|$. Moreover, $\mathbf{x} := \operatorname{col}(\mathbf{x}_1, ..., \mathbf{x}_n)$ is equal to $\mathbf{x} := (\mathbf{x}_1^\top, ..., \mathbf{x}_n^\top)^\top$. A set-value mapping $\mathcal{M} : \mathcal{A} \rightrightarrows \mathcal{B}$ is such that $\mathcal{A} \mapsto 2^{\mathcal{B}}$, for some sets $\mathcal{A}, \mathcal{B}$, where $2^{\mathcal{B}}$ is the power set of $\mathcal{B}$. We define the mapping $\operatorname{proj}_{\mathcal{X}} : \mathbb{R}^n \to \mathcal{X}$ as the projection into the generic closed non-empty set $\mathcal{X} \subseteq \mathbb{R}^n$, i.e., $\operatorname{proj}_{\mathcal{X}}(\mathbf{y}) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{y}\|$. A possibly (nonlinear) mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is said to be Lipschitz continuous with a constant $\ell \in \mathbb{R}_{>0}$ if $\|F(\mathbf{x}) - F(\mathbf{y})\| \leq \ell \|\mathbf{x} - \mathbf{y}\|$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. Given two mappings $F : \mathcal{X} \to \mathcal{Y}$ and : $\mathcal{Y} \to \mathcal{Z}$, their composition $H : \mathcal{X} \to \mathcal{Z}$ such that $H(\cdot) = F(G(\cdot))$ is denoted as $H(\cdot) = F \circ G(\cdot)$.

B. Preliminaries on Game Theory

Let us consider the standard mathematical setting of noncooperative games [21]. Thus, let us consider a set of N agents $\mathcal{N}$, indexed by $i \in \mathcal{N} := \{1, ..., N\} \subseteq \mathbb{N}$ each with decision variables $\mathbf{x}_i \in \mathbb{R}^{n_i}$, for some $n_i \in \mathbb{N}$. Moreover, let $n := \sum_{i\in\mathcal{N}} n_i$. We define vector $\mathbf{x}_{-i} := \operatorname{col}(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_N) \in \mathbb{R}^{n_{-i}}$, where $n_{-i} := n - n_i$, which collects the strategies of all agents but i, as well as vector $\mathbf{x} := \operatorname{col}(\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N) \in \mathbb{R}^n$, collecting the strategy of all agents. Each agent $i \in \mathcal{N}$ tries to minimize a (possibly non-convex) cost function $f_i(\mathbf{x}_i, \mathbf{x}_{-i}) : \mathbb{R}^n \times \mathbb{R}^{n-n_i} \to \mathbb{R}$ by choosing a strategy in a (possibly non-convex) feasible set $\mathbf{x}_i \in \Omega_i \subseteq \mathbb{R}^n$. Moreover, let $\mathbf{x} \in \Omega = \prod_{i\in\mathcal{N}} \Omega_i$. One can thus define the so-called Nash equilibrium problem (NEP) as the following N interdependent optimization problem:

$$\forall i \in \mathcal{N} : \quad \underset{\mathbf{x}_i\in\Omega_i}{\operatorname{minimize}}\ f_i(\mathbf{x}_i, \mathbf{x}_{-i}). \tag{1}$$

A solution for (1) is a Nash equilibrium (NE), formally defined as follows.

Definition 1 (Nash equilibrium)**.** *A NE is a collective strategy $\mathbf{x}^\bullet \in \mathcal{X}$ such that:*

$$\forall i \in \mathcal{N} : \quad f_i(\mathbf{x}_i^\bullet, \mathbf{x}_{-i}^\bullet) \leq \inf\left\{f_i(\mathbf{x}_i, \mathbf{x}_{-i}^\bullet) \,|\, \mathbf{x}_i \in \Omega_i\right\}. \tag{2}$$

In other words, a NE is a collective strategy profile that satisfies the property that no single agent in the game can improve its objective function by unilaterally changing its strategy to another feasible one.

A standard requirement, often introduced in related works, is that cost functions are convex, or at least quasi-convex, with respect to their own strategy. Therefore, we lay out the following Assumption.

Assumption 1. *For each $i \in \mathcal{N}$ and for every $\mathbf{x}_{-i}$, the function $f_i(\cdot, \mathbf{x}_{-i})$ is convex and continuously differentiable.*

Assumption 2. *The set Ω is convex and compact.*

For instance, Assumptions 1 and 2 are necessary for setting up many fixed-point formulations used to demonstrate the existence and convergence of Nash equilibria [22]. Notably, equilibrium existence may break without (quasi-)convexity, even for very simple games [23].

C. Preliminaries on Neural Networks

We consider a feed-forward DNN, thus a network where information moves in only one direction with no cycles or loops [24]. Each layer $l \in \mathcal{L} := \{1, ..., L\} \subseteq \mathbb{N}$ is a processing ensemble comprising a set of neurons $\mathcal{P}_l$. The output of each layer $\mathbf{x}_l \in \mathbb{R}^{|\mathcal{P}_l|}$ can be computed as:

$$\mathbf{x}_l = \Phi_l\left(\mathbf{W}_l \mathbf{x}_{l-1} + \mathbf{b}_l\right), \quad \forall l \in \mathcal{L} \tag{3}$$

where $\mathbf{W}_l \in \mathbb{R}^{|\mathcal{P}_l|\times|\mathcal{P}_{l-1}|}$ is the weight matrix, $\mathbf{b}_l \in \mathbb{R}^{|\mathcal{P}_l|}$ the bias vector and $\Phi_l : \mathbb{R}^{|\mathcal{P}_l|} \to \mathbb{R}^{|\mathcal{P}_l|}$ the activation function of the layer. The weights and biases are the parameters that define the function approximator, and their values are identified through a data-driven optimization process known as training [25]. During training, weights and biases are modified to minimize a loss function, typically representing the discrepancy between the predicted outputs and the actual targets in a given dataset.

By setting $\mathbf{x}_0 \in \mathbb{R}^{|\mathcal{P}_0|}$ as the input and $\mathbf{x}_L \in \mathbb{R}^{|\mathcal{P}_L|}$ as the output of the DNN, we can define the overall input-output relationship of the network $\Phi : \mathbb{R}^{|\mathcal{P}_0|} \to \mathbb{R}^{|\mathcal{P}_L|}$ in the following form:

$$\mathbf{x}_L = \Phi\left(\mathbf{x}_0\right) \tag{4}$$

where $\Phi(\cdot) = \Phi_L \circ \Phi_{L-1} \circ \cdots \circ \Phi_1(\cdot)$.

Let us restrict our attention to a specific class of layers and activation functions.

Assumption 3. *Assume that the following properties hold:*

3.1 Given $\mathbf{y}_l, \mathbf{z}_l \in \mathbb{R}^{|\mathcal{P}_l|}$, there exists a $\gamma_l \in \mathbb{R}_{\geq 0}$ such that:

$$\begin{aligned}\|\left(\mathbf{W}_l \mathbf{y}_{l-1} + \mathbf{b}_l\right) - \left(\mathbf{W}_l \mathbf{z}_{l-1} + \mathbf{b}_l\right)\| \\ \leq \gamma_l \|\mathbf{y}_{l-1} - \mathbf{z}_{l-1}\|, \quad \forall l \in \mathcal{L}.\end{aligned} \tag{5}$$

3.2 Given $\mathbf{y}_l, \mathbf{z}_l \in \mathbb{R}^{|\mathcal{P}_l|}$:

$$\|\Phi_l\left(\mathbf{y}_{l-1}\right) - \Phi_l\left(\mathbf{z}_{l-1}\right)\| \leq \|\mathbf{y}_{l-1} - \mathbf{z}_{l-1}\|, \quad \forall l \in \mathcal{L}. \tag{6}$$

It follows from Assumption 3.1 that

$$\begin{aligned}\|\left(\mathbf{W}_l \mathbf{y}_{l-1} + \mathbf{b}_l\right) - \left(\mathbf{W}_l \mathbf{z}_{l-1} + \mathbf{b}_l\right)\| &= \|\mathbf{W}_l(\mathbf{y}_{l-1} - \mathbf{z}_{l-1})\| \\ &= \|\mathbf{y}_{l-1} - \mathbf{z}_{l-1}\|_{\mathbf{H}_l} \\ &\leq \gamma_l \|\mathbf{y}_{l-1} - \mathbf{z}_{l-1}\|\end{aligned}$$

where $\mathbf{H}_l := \mathbf{W}_l^\top \mathbf{W}_l \succeq 0$. By rearranging the terms in the last inequality, we obtain

$$\begin{aligned}\gamma_l \|\mathbf{y}_{l-1} - \mathbf{z}_{l-1}\| - \|\mathbf{y}_{l-1} - \mathbf{z}_{l-1}\|_{\mathbf{H}_l} &= \|\mathbf{y}_{l-1} - \mathbf{z}_{l-1}\|_{\gamma_l \mathbf{I} - \mathbf{H}_l} \\ &\geq 0\end{aligned}$$

which holds iff $\gamma_l \mathbf{I} - \mathbf{H}_l \succeq 0$. Albeit no requirements are needed for the biases, the last inequality forces the training process into a semidefinite optimization problem, with the non-convex constraint $\mathbf{W}_l^\top \mathbf{W}_l \preceq \gamma_l \mathbf{I}$. A naive idea to overcome such issue is to evaluate $\tilde{\mathbf{H}}_l = \mathbf{H}_l - \mathbf{H}_l \mathbf{1} - \gamma_l \mathbf{I} \preceq 0$, where the latter holds from Gershgorin circle theorem, and then recovering $\tilde{\mathbf{W}}_l^\top \tilde{\mathbf{W}}_l = \tilde{\mathbf{H}}_l$ through Cholesky decomposition. An in-depth discussion on the properties of Lipschitz neural network is provided in [26].

Additionally, it is trivial to show that the most commonly used activation functions, such as sigmoid, hyperbolic tangent, and rectified linear unit (ReLU), satisfy Assumption 3.2. In contrast, the Gaussian activation function does not satisfy it.

Lemma 1. *A feed-forward DNN* (4) *respecting Assumption 3 is a Lipschitz continuous map with constant* $\gamma = \prod_{l \in \mathcal{L}} \gamma_l$.

Proof A feed-forward DNN is essentially a stack of layers, where each layer transforms the previous layer's output and feeds its output to the next ones. By the composition property of Lipschitz functions the map $\Phi(\cdot) = \Phi_L \circ \Phi_{L-1} \circ \cdots \circ \Phi_1$ is Lipschitz continuous with a constant $\gamma = \prod_{l \in \mathcal{L}} \gamma_l$, being γ_l the Lipschitz constant of the layer $l \in \mathcal{L}$. ■

III. Learning-Based Nash Games

Games are typically formulated, as in (1), by approximating agents' preferences using cost functions which ultimately drive the agent's behavior. Thus, the applicability of such multi-agent models is limited by (i) the possibility of finding suitable functions that realistically model agents' preferences and (ii) constraining their formulation to be convex, as per Assumption 1. As a result, these limitations do not allow for modeling scenarios where agents' behaviors are extremely complex.

Despite the challenging task of defining a function that realistically approximates agents' preferences, several applications allow for measuring agents' behavior in the sense of evaluating their response to a given environment whose state depends on given parameters and other agents' decisions.

Let $\mathcal{E} := \mathcal{S} \times \Omega$ abstract an environment whose state is (uniquely) determined by a certain scenario $\mathbf{s} \in \mathcal{S} \subset \mathbb{R}^m$, for some $m \in \mathbb{N}$, and the collective actions taken by the agents $\mathbf{x} \in \Omega$. From the perspective of agent $i \in \mathcal{N}$, however, only set $\mathcal{E}_i := \mathcal{S}_i \times \Omega_{-i}$ is accessible, where $\mathcal{S}_i \subset \mathbb{R}^{q_i}$, with $q_i \leq m$ (possibly strictly), acting as the co-domain of some $g_i : \mathcal{S} \to \mathcal{S}_i$, represents the information agent $i \in \mathcal{N}$ acquires regarding scenario $\mathbf{s} \in \mathcal{S}$. Moreover, let $\Omega_{-i} := \Omega_1 \cap \cdots \cap \Omega_{i-1} \cap \Omega_{i+1} \cap \cdots \cap \Omega_N$ constitute the set of all strategies but the one of player $i \in \mathcal{N}$. Formally, we can introduce an observer $O_i : \mathcal{E} \to \mathcal{E}_i$ which maps $(\mathbf{s}, \mathbf{x}) \mapsto \mathrm{col}(g_i(\mathbf{s}), h_i(\mathbf{x}_{-i}))$ for some $h_i : \Omega_{-i} \to \mathbb{R}^{p_i}$, with $p_i \leq n - n_i$ (possibly strictly). The latter represents the information quota that agent i receives regarding other agents strategies. A scheme of this framework is shown in Fig. 1. Therefore, each agent has a possibly limited view of the environment, albeit a full knowledge of some function of the other's decisions. In full-information games, one has $h(\mathbf{x}_{-i}) = \mathbf{x}_{-i}$. Similarly, popular equilibrium-seeking algorithms follow such a setup, e.g., in the class of aggregative games, we have $h_i(\mathbf{x}_{-i}) = \sum_{j \in \mathcal{N} \setminus \{i\}} x_j$.

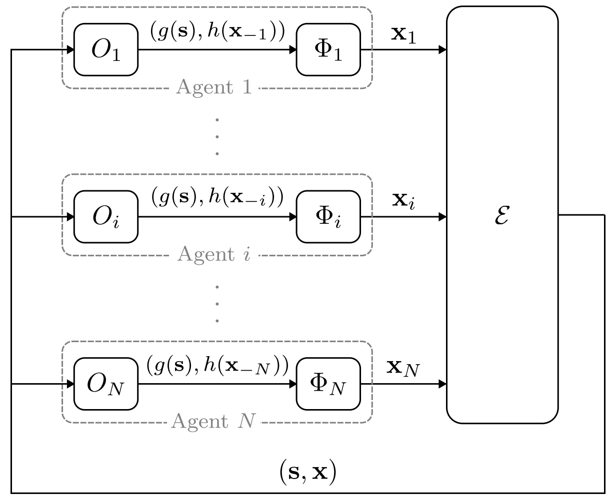

Fig. 1. Scheme of the proposed learning-based noncooperative game framework.

Such a formulation allows us to define the training set and target vector against which each agent $i \in \mathcal{N}$ can develop its response strategy. Specifically, we assume that each agent $i \in \mathcal{N}$ can access a training set $\mathcal{T}_i \subset \mathcal{E}_i \times \Omega_i$, so that a tuple $(O_i(\mathbf{s}, \mathbf{x}), \mathbf{x}_i) \in \mathcal{T}_i$ is such that each the environmental input $O_i(\mathbf{s}, \mathbf{x})$ determines a response action $\mathbf{x}_i \in \Omega_i$. A note of caution should be used in defining the nature of the response $\mathbf{x}_i$: due to the lack of an index capable of introducing a preference relation on Ω_i, the action taken by each agent, as a response to environmental stimuli does not necessarily classify as "optimal". Therefore, such a setup bears weaker assumptions on the nature of $\mathbf{x}_i$, with respect to the most commonly used best-response formulations, where each agent reacts to others' actions by iteratively solving (1). This allows for modeling agents with limited rationality and subject to environmental conditioning. Equipped with a dataset $\mathcal{T}_i$, we can define a feed-forward DNN as in (4), and train it to approximate the behavior of agent $i \in \mathcal{N}$ as:

$$\forall i \in \mathcal{N}: \quad \tilde{\mathbf{x}}_i = \Phi_i\left(\begin{bmatrix} g_i(\mathbf{s}) \\ h_i(\mathbf{x}_{-i}) \end{bmatrix}\right) = \Phi_i(O_i(\mathbf{s}, \mathbf{x})) \tag{7}$$

where $\tilde{\mathbf{x}}_i$ is the response of agent $i \in \mathcal{N}$ yielded by the feed-forward DNN when the actions of other agents are $\mathbf{x}_{-i} \in \mathbb{R}^{n-n_i}$ under scenario $\mathbf{s} \in \mathcal{S}$. Note that, as the strategy returned by (7) may yield infeasible as is provided by a feed-forward DNN, i.e., $\tilde{\mathbf{x}}_i \notin \Omega_i$. Thus, let us project into the feasible set Ω_i the DNN output as:

$$\forall i \in \mathcal{N}: \quad \mathbf{x}_i = \mathrm{proj}_{\Omega_i} \{\Phi_i(O_i(\mathbf{s}, \mathbf{x}))\}. \tag{8}$$

Remark 1. *Once trained,* (8) *becomes an alternative approach to evaluating* $\mathbf{x}_i = \mathrm{argmin}_{\mathbf{x}_i \in \Omega_i} f_i(\mathbf{x}_i, \mathbf{x}_{-i})$*, as in* (1)*, when deriving suitable convex objectives is inconvenient.*

IV. Existence and Uniqueness of Equilibria

Having redefined agents' behavior, the standard setting for Nash equilibria does not hold here. Thus, let us search for different equilibrium conditions and introduce a notion of *Learning-Based Nash Equilibrium* (LBNE), defined as follows.

Definition 2 (Learning-Based Nash Equilibrium)**.** *A LBNE is a strategy profile $\mathbf{x}^* \in \Omega$ such that, for any $\mathbf{s} \in \mathcal{S}$:*

$$\forall i \in \mathcal{N}: \quad \mathbf{x}_i^* = \mathrm{proj}_{\Omega_i} \left\{ \Phi_i(O_i(\mathbf{s}, \mathbf{x}^*)) \right\}. \tag{9}$$

Intuitively, a LBNE comprises strategies satisfying no specific optimality condition, as agents essentially make their moves in response to their opponents' strategies based on the results of a data-based approach, which can lead to suboptimal solutions when compared with the results yielded by an ideal $f_i(\cdot, \cdot)$.

Next, we argue that an LBNE equilibrium exists under the following assumptions.

Assumption 4. *For each $i \in \mathcal{N}$ the feed-forward DNN $\Phi_i(\cdot)$, approximating the agent response, is Lipschitz continuous with constant γ_i, while $h_i : \Omega_{-i} \to \mathbb{R}^{p_i}$ is 1-Lipschitz continuous.*

Note that requiring the feed-forward DNN $\Phi_i(\cdot)$ to be Lipschitz continuous with constant γ_i is equivalent to requiring that it is composed of a set of layers $\mathcal{L}_i$, such that for each layer $l_i \in \mathcal{L}_i$, Assumption 3 holds with a constant $\gamma_{l,i}$. This requirement ensures that $\gamma_i := \prod_{l \in \mathcal{L}_i} \gamma_{l,i}$, as specified in Lemma 1.

Proposition 1 (Existence)**.** *Every game satisfying Assumptions 2 and 4, has at least one LBNE.*

Proof Consider the mapping $M : \mathbb{R}^{Nn} \to \mathbb{R}^{Nn}$ defined as follows:

$$M : \mathbf{x} \mapsto \mathrm{proj}_{\Omega} \begin{pmatrix} \Phi_1(O_1(\mathbf{s}, \mathbf{x})) \\ \vdots \\ \Phi_N(O_N(\mathbf{s}, \mathbf{x})) \end{pmatrix}, \tag{10}$$

as the observer is a nonexpansive, the feed-forward DNNs $\Phi_i(\cdot)$ are γ_i-Lipschitz continuous due to Assumption 3, and, since the projection to a convex and compact set Ω is continuous, the map $M(\cdot)$ is γ_m-Lipschitz continuous [27].

Given the continuity of $M(\cdot)$ and the convexity and compactness of Ω, it follows from Brouwers fixed point theorem that $M(\cdot)$ has a fixed point $\mathbf{x}^* = M(\mathbf{x}^*)$. We will argue that (9) holds for the fixed point $\mathbf{x}^*$. Indeed, denoting by:

$$\mathbf{z}^* = \begin{pmatrix} \Phi_1(O_1(\mathbf{s}, \mathbf{x}^*)) \\ \vdots \\ \Phi_N(O_N(\mathbf{s}, \mathbf{x}^*)) \end{pmatrix} \tag{11}$$

and since $\mathbf{x}^* = \mathrm{proj}_{\Omega}(\mathbf{z}^*)$ and, due to a well-known propriety of projection, we have that:

$$(\mathbf{z}^* - \mathbf{x}^*)^\top (\mathbf{x}^* - \mathbf{y}) \geq 0, \quad \forall \mathbf{y} \in \Omega \tag{12}$$

for an arbitrary $\mathbf{x}_i \in \Omega_i$, setting $\mathbf{y} = (\mathbf{x}_i, \mathbf{x}_{-i}^*)$ into the above inequality we get that:

$$(\Phi_i(O_i(\mathbf{s}, \mathbf{x}^*)) - \mathbf{x}_i^*)^\top (\mathbf{x}_i^* - \mathbf{x}_i) \geq 0 \tag{13}$$

Algorithm 1 Picard-Banach Distributed Scheme

```
Set x_i^0 ∈ Ω_i, ∀i ∈ N
for k = 0, ..., ∞ do
    for i ∈ N do
        x_i^{k+1} ← proj_{Ω_i}(Φ_i(O_i(s, x^k))).
    end for
end for
```

thus we have:

$$(\Phi_i(O_i(\mathbf{s}, \mathbf{x}^*)) - \mathbf{x}_i^*)^\top (\mathbf{x}_i^* - \mathbf{x}_i) \geq 0, \quad \forall \mathbf{x}_i \in \Omega_i \tag{14}$$

which is equivalent to (9). ■

Proposition 2 (Uniqueness)**.** *Every game satisfying Assumptions 2 and 4 with $\left(\sum_{i \in \mathcal{N}} \gamma_i^2\right)^{1/2} < 1$ has only one LBNE.*

Proof If $\left(\sum_{i \in \mathcal{N}} \gamma_i^2\right)^{1/2} < 1$, the mapping (10) is a contraction that has a unique fixed point [28], which is a LBNE due to Proposition 1. ■

V. Convergence to an Equilibria

In this Section, let us present two distributed LBNE-seeking approaches. Intuitively, the mapping $M(\cdot)$ characterizes what would happen if all agents were to synchronously compute and update their strategies based on other agents' decisions. We are thus interested in characterizing the asymptotic properties of the response evolution when this step is repeated indefinitely, as described in Algorithm 1, given an initial state $\mathbf{x}_i^0 \in \Omega_i$ for all agents.

Proposition 3. *Suppose that Assumptions 2 and 3 hold with $\left(\sum_{i \in \mathcal{N}} \gamma_i^2\right)^{1/2} < 1$. Then, for an initial state $\mathbf{x}_i^0 \in \Omega_i$ for all agents $i \in \mathcal{N}$ the sequence $(\mathbf{x}^k)_{k=0}^{\infty}$ converges to the unique fixed point of* (10)*. Thus, by Propositions 1 and 2 the sequence $(\mathbf{x}_i^k)_{k=0}^{\infty}$ converges to the unique LBNE* (9)*.*

Proof Since $\left(\sum_{i \in \mathcal{N}} \gamma_i^2\right)^{1/2} < 1$, the mapping $M(\cdot)$ is a contraction and thus converges, for any initial condition $\mathbf{x}^{(0)} \in \mathbb{R}^n$, to its unique fixed point [29], [30], which is a LBNE due to Proposition 1. ■

The conditions under which Algorithm 1 converges may be too restrictive in some cases, such as the contractiveness of the mapping $M(\cdot)$. Indeed, relaxing this latter assumption and requiring the mapping $M(\cdot)$ to be nonexpansive only is insufficient for the PicardBanach iteration to converge to a fixed point. Thus, let us assume here that agents compute their response with a convex combination between the current other agents' strategies and the response used at the previous iteration, that is, the well-known Krasnoselskij iteration, described in Algorithm 2.

Proposition 4. *Suppose that Assumptions 2 and 3 hold with $\left(\sum_{i \in \mathcal{N}} \gamma_i^2\right)^{1/2} = 1$. Then for an initial state $\mathbf{x}_i^0 \in \Omega_i$ for all agents $i \in \mathcal{N}$ the sequence $(\mathbf{x}^k)_{k=0}^{\infty}$ converges to a fixed point of* (10)*. Thus, by Proposition 1 the sequence $(\mathbf{x}_i^k)_{k=0}^{\infty}$ converges for to a LBNE* (9)*.*

Algorithm 2 Krasnoselskij Distributed Scheme

1: Set $\mathbf{x}_i^0 \in \Omega_i, \forall i \in \mathcal{N}$
2: **for** $k = 0, \dots, \infty$ **do**
3: **for** $i \in \mathcal{N}$ **do**
4: $\mathbf{x}_i^{k+1} \leftarrow (1-\alpha)\mathbf{x}_i^k + \alpha \operatorname{proj}_{\Omega_i}(\Phi_i(O_i(\mathbf{s}, \mathbf{x}^k))$.
5: **end for**
6: **end for**

Proof Since $\left(\sum_{i \in \mathcal{N}} \gamma_i^2\right)^{1/2} = 1$, the mapping $M(\cdot)$ is a nonexpansive mapping thus converges, for any initial condition $x^{(0)} \in \mathbb{R}^n$, to a unique fixed point [29], [30], which is a LBNE due to Proposition 1. ■

VI. Illustrative Example

As an illustrative example, let us consider an energy community model comprising smart energy users. Each agent $i \in \mathcal{N}$ behaves selfishly, choosing its energy consumption strategy $\mathbf{x}_i$ from a convex and compact feasible set Ω_i, i.e., $\mathbf{x}_i \in \Omega_i$.

For ease of presentation, let us assume that the energy cost in the community follows a dynamic pricing scheme, where the cost incurred by agent $i \in \mathcal{N}$ depends on the strategies of other agents $\mathbf{x}_{-i}$ [31]. This setting allows us to present our results in an aggregative fashion. Nevertheless, note that the results presented in this section hold for generally coupled games respecting Assumption 4. This allows us to present our results in an aggregative fashion. However, it's important to note that the results presented in this section are applicable to generally coupled games that adhere to Assumption 4.

Specifically, we assume that the energy cost for each consumer is an aggregation of other agents' strategies and thus can be computed as:

$$h_i(\mathbf{x}_{-i}) := \sum_{j \in \mathcal{N} \setminus \{i\}} P_{i,j} \mathbf{x}_j \tag{15}$$

where $P_{i,j}$ indicates the strength of the influence of agent $j \in \mathcal{N}$ on agent $i \in \mathcal{N}$, with 0 denoting no influence. Specifically, we assume $P := \frac{1}{N-1}(\mathbf{1}_N \mathbf{1}_N^\top - I_N)$, which is doubly stochastic satisfying condition $\|P\| \leq 1$. Note that since $\|P\| \leq 1$ Assumption 4 hold for $h_i(i)$ [27].

To train the DNNs approximating agents' behavior, we utilize data from the *Low Carbon London* project [32]. This dataset comprises energy consumption readings for 5,567 households in London, collected between November 2011 and February 2014 at half-hourly intervals. The households were selected to represent a balanced sample of the Greater London population.

The dataset includes energy consumption in kWh per half hour, unique household identifiers, dates, and times for approximately 1,100 customers subjected to dynamic energy prices. Tariff prices were provided a day in advance through Smart Meter IHDs or text messages to mobile phones. Both the date/time information and the price signal schedule are available in the dataset. Some analysis of this dataset is available here [33].

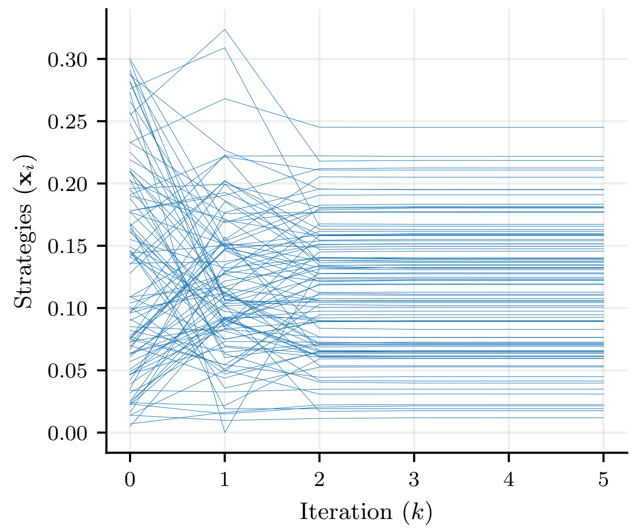

Fig. 2. Strategy convergence of agents using Algorithm 1.

For the training process, we select 100 customers from the dataset. We train a different DNN for each customer using timestamp information and the corresponding energy price as input features. The target value for each DNN is the respective customer's energy consumption. Thus, each DNN aims to approximate a customer's behavior in deciding how much energy to use in a specific time slot based on the energy price.

The DNNs are trained using the DEEPLip library, which is specifically designed to train Lipschitz layers [34]. In particular, we employ DNNs composed of 5 linear layers with 64 neurons each and fullsort activation functions.

In Fig. 2, we show the convergence of strategies to an LBNE using Algorithm 1, while in Fig. 3, the results of Algorithm 2 with a parameter $\alpha = 0.3$. Please note that the equilibrium reached by both algorithms is the same, as all the DNNs are Lipschitz continuous with $\gamma_i < 1$, ensuring the uniqueness of the LBNE. From the comparison of the two algorithms, it emerges that the average number of iterations required by Algorithm 1 is lower.

VII. Conclusion

This paper challenges the conventional reliance on convexity in game theory, recognizing the limitations it imposes when agents' utility functions cannot be adequately represented preserving this assumption. Unlike conventional approaches that model agents' behaviors with convex cost functions, we propose using deep neural networks (DNNs) to compute the agents' response actions. Introducing a technical assumption on parameters of the DNN, we establish the existence and uniqueness of equilibria. Two distributed algorithms based on fixed-point iterations are presented for their computation, showing the practicality of our approach.

As a future work, it would be interesting to extend the proposed framework to games with coupling constraints.

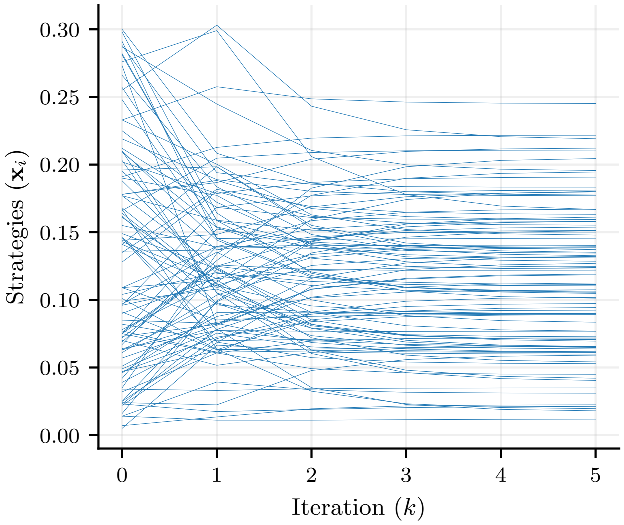

Fig. 3. Strategy convergence of agents using Algorithm 2.

REFERENCES

[1] R. B. Myerson, "Nash equilibrium and the history of economic theory," *Journal of Economic Literature*, vol. 37, no. 3, pp. 1067–1082, 1999.

[2] J. Von Neumann, "Zur theorie der gesellschaftsspiele, in «mathematische annalen», 100," 1928.

[3] B. Brooks and P. J. Reny, "A canonical game–nearly 75 years in the making–showing the equivalence of matrix games and linear programming," *Available at SSRN 3851583*, 2021.

[4] N. Mignoni, P. Scarabaggio, R. Carli, and M. Dotoli, "Control frameworks for transactive energy storage services in energy communities," *Control Engineering Practice*, vol. 130, p. 105364, 2023.

[5] E. Bakolas and Y. Lee, "Decentralized game-theoretic control for dynamic task allocation problems for multi-agent systems," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3228–3233.

[6] P. Aditya and H. Werner, "A distributed linear-quadratic discrete-time game approach to multi-agent consensus," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 6169–6174.

[7] T. R. Oliveira, V. H. P. Rodrigues, M. Krstić, and T. Baar, "Nash equilibrium seeking in heterogeneous noncooperative games with players acting through heat pde dynamics and delays," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 1167–1173.

[8] P. Scarabaggio, S. Grammatico, R. Carli, and M. Dotoli, "Distributed demand side management with stochastic wind power forecasting," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 97–112, 2021.

[9] X. Huang, B. Beferull-Lozano, and C. Botella, "Quasi-nash equilibria for non-convex distributed power allocation games in cognitive radios," *IEEE Transactions on Wireless Communications*, vol. 12, no. 7, pp. 3326–3337, 2013.

[10] P. Scarabaggio, R. Carli, and M. Dotoli, "Noncooperative equilibrium-seeking in distributed energy systems under ac power flow nonlinear constraints," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 4, pp. 1731–1742, 2022.

[11] C. Biasi, T. F. M. Monis *et al.*, "Weak local Nash equilibrium," *Topological Methods in Nonlinear Analysis*, vol. 41, no. 2, pp. 409–419, 2013.

[12] C. Alós-Ferrer and A. B. Ania, "Local equilibria in economic games," *Economics Letters*, vol. 70, no. 2, pp. 165–173, 2001.

[14] M. R. Baye, G. Tian, and J. Zhou, "Characterizations of the existence of equilibria in games with discontinuous and non-quasiconcave payoffs," *The Review of Economic Studies*, vol. 60, no. 4, pp. 935–948, 1993.

[13] B. Cornet and M.-O. Czarnecki, "Existence of generalized equilibria," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 44, no. 5, pp. 555–574, 2001.

[15] C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan, "On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points," *Journal of the ACM (JACM)*, vol. 68, no. 2, pp. 1–29, 2021.

[16] L. Deng and X. Li, "Machine learning paradigms for speech recognition: An overview," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 1060–1089, 2013.

[17] A. Maier, C. Syben, T. Lasser, and C. Riess, "A gentle introduction to deep learning in medical image processing," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 86–101, 2019.

[18] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (gans): A survey," *IEEE access*, vol. 7, pp. 36 322–36 333, 2019.

[19] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

[20] A. Rosenfeld and S. Kraus, "Predicting human decision-making," in *Predicting Human Decision-Making: From Prediction to Action*. Springer, 2018, pp. 21–59.

[21] C. De Persis and S. Grammatico, "Distributed averaging integral nash equilibrium seeking on networks," *Automatica*, vol. 110, p. 108548, 2019.

[22] J. F. Nash Jr, "Equilibrium points in n-person games," *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[23] C. Daskalakis, N. Golowich, S. Skoulakis, and E. Zampetakis, "Stay-on-the-ridge: Guaranteed convergence to local minimax equilibrium in nonconvex-nonconcave games," in *The Thirty Sixth Annual Conference on Learning Theory*. PMLR, 2023, pp. 5146–5198.

[24] J. Seo, J. Lee, and K. Kim, "Decoding of polar code by using deep feed-forward neural networks," in *2018 international conference on computing, networking and communications (ICNC)*. IEEE, 2018, pp. 238–242.

[25] B. Solarte, C.-H. Wu, Y.-C. Liu, Y.-H. Tsai, and M. Sun, "360-mlc: Multi-view layout consistency for self-training and hyper-parameter tuning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6133–6146, 2022.

[26] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: analysis and efficient estimation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[27] K. Eriksson, C. Johnson, D. Estep, K. Eriksson, C. Johnson, and D. Estep, "Vector-valued functions of several real variables," *Applied Mathematics: Body and Soul: Calculus in Several Dimensions*, pp. 789–814, 2004.

[28] V. Berinde and M. Pacurar, "Fixed points and continuity of almost contractions," *Fixed Point Theory*, vol. 9, no. 1, pp. 23–34, 2008.

[29] H. Bauschke and P. Combettes, "Convex analysis and monotone operator theory in hilbert spaces, corrected printing," 2019.

[30] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.

[31] D. Paccagnan, B. Gentile, F. Parise, M. Kamgarpour, and J. Lygeros, "Distributed computation of generalized nash equilibria in quadratic aggregative games with affine coupling constraints," in *2016 IEEE 55th conference on decision and control (CDC)*. IEEE, 2016, pp. 6123–6128.

[32] "Low Carbon London - UKPN Innovation," Oct. 2023, [Online; accessed 21. Mar. 2024]. [Online]. Available: https://innovation.ukpowernetworks.co.uk/projects/low-carbon-london

[33] "Electricity Consumption in a Sample of London Households – London Datastore," Mar. 2024, [Online; accessed 21. Mar. 2024]. [Online]. Available: https://data.london.gov.uk/blog/electricity-consumption-in-a-sample-of-london-households

[34] M. Serrurier, F. Mamalet, A. González-Sanz, T. Boissin, J.-M. Loubes, and E. Del Barrio, "Achieving robustness in classification using optimal transport with hinge regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 505–514.