Rationality of Learning Algorithms in Repeated Normal-Form Games

Shivam Bajaj, *Member, IEEE*, Pranoy Das, Yevgeniy Vorobeychik, and Vijay Gupta, *Fellow, IEEE*

***Abstract*—Many learning algorithms are known to converge to an equilibrium for specific classes of games if the same learning algorithm is adopted by all agents. However, when the agents are self-interested, a natural question is whether the agents have an incentive to unilaterally shift to an alternative learning algorithm. We capture such incentives as an algorithm's *rationality ratio*, which is the ratio of the highest payoff an agent can obtain by unilaterally deviating from a learning algorithm to its payoff from following it. We define a learning algorithm to be *c-rational* if its rationality ratio is at most c irrespective of the game. We show that popular learning algorithms such as fictitious play and regret-matching are not c-rational for any constant $c \geq 1$. We also show that if an agent can only observe the actions of the other agents but not their payoffs, then there are games for which c-rational algorithms do not exist. We then propose a framework that can build upon any existing learning algorithm and establish, under mild assumptions, that our proposed algorithm is (i) c-rational for a given $c \geq 1$ and (ii) the strategies of the agents converge to an equilibrium, with high probability, if all agents follow it.**

***Index Terms*—Game theory, learning in games, agents-based systems.**

I. Introduction

USE OF automated learning agents is increasing in various online applications such as automated trading and online auctions. Intuitively, these agents play a game repeatedly and update their strategies to converge to some equilibrium concept. Various multi-agent learning algorithms with desirable convergence properties have been proposed, such as fictitious play, regret-matching, gradient descent, etc. [1], [2], [3]. A natural question in such situations is whether any individual agent has an incentive to rewrite its algorithm unilaterally to increase its payoff. Recent works have considered this question and have shown that a strategic agent can indeed exploit the knowledge of the underlying learning algorithm to increase its payoff [4], [5], [6]. Intuitively, this is because the learning algorithms themselves do not constitute equilibrium behavior of the corresponding repeated game. In response, [7] introduced the notion of a *learning equilibrium* that requires that the learning algorithms themselves be *rational* in that self-play for that algorithm is a symmetric equilibrium of the repeated game and proposed algorithms that exhibit this property. However, [7] assumes a uniform bound on game payoffs and does not evaluate whether conventional learning approaches, such as fictitious play or regret-matching, are already (nearly) in equilibrium in self-play and if not, whether we can induce this property in a way that is outcome-equivalent to these algorithms in self-play. In a related work, [8] proposed an algorithm that is *non-exploitable* in the sense that it ensures a payoff above a certain value if the other agent deviates from the algorithm. However, it restricts the strategies of the deviating agent.

Designing learning algorithms that are rational in self-play is challenging since such algorithms may not even exist for certain classes of games [7]. Thus, a quantifiable metric that characterizes how much an agent may benefit by deviating from its learning algorithm is required to allow the system designer to compare various algorithms and choose one.

To this end, we consider a two-agent repeated game framework and introduce the concept of *rationality ratio*, defined as the ratio of the most an agent can obtain by unilaterally deviating from a learning algorithm to their payoff from following it. For a constant $c \geq 1$, a learning algorithm is c-rational if its rationality ratio is no more than c in the worst-case. We first show that there does not exist any constant $c \geq 1$ for which classic learning algorithms such as fictitious play and regret-matching are c-rational. We also establish that there exist games for which c-rational algorithms do not exist if an assumption of *perfect monitoring* does not hold. We then design and analyze an algorithm that builds on any existing learning algorithm and, under mild assumptions, is provably c-rational, for a given c, while converging to the same equilibrium as the underlying algorithm. Similar to [7], [8], to deter an agent from deviating from the specified learning algorithms, we utilize an approach by which the agents *punish* this deviating agent. Such punishment strategies are acceptable since for automated agents following a prescribed algorithm, strategic manipulation is most salient ex-ante and the primary

Received 13 August 2024; revised 7 October 2024; accepted 19 October 2024. Date of publication 25 October 2024; date of current version 1 November 2024. This work was supported in part by the Army Research Office (ARO) under Grant W911NF2310111; in part by the Office of Naval Research (ONR) under Grant F.10052139.02.012 and Grant N00014-24-1-2663; and in part by NSF under Grant IIS-1905558 and Grant IIS-2214141. Recommended by Senior Editor M. Guay. *(Corresponding author: Shivam Bajaj.)*

Shivam Bajaj, Pranoy Das, and Vijay Gupta are with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: bajaj41@purdue.edu; das211@purdue.edu; gupta869@purdue.edu).

Yevgeniy Vorobeychik is with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: yvorobeychik@wustl.edu).

Digital Object Identifier 10.1109/LCSYS.2024.3486631

requirement is to avoid creating incentives for the owner of such agents to rewrite the learning algorithms being used.

Our work is related to the growing area of strategizing against learning agents [9], [10]. These works primarily focus on optimization approaches for a single agent whereas we consider a game theoretic setting. Another closely related line of work considers *rational learning* in which the agents best-responds to the belief over the strategies of the other agent [11]. However, it has been argued that rational learning does not converge to Nash equilibrium in general [12]. Unlike [11] our agents do not use Bayesian belief updates.

This letter is organized as follows. Section II formally describes the model. Section III establishes that some classical learning algorithms are not c-rational and Section IV establishes non-existence of any c-rational algorithm if a perfect monitoring assumption does not hold. Section V presents our algorithm that is provably c-rational for a given c and Section VI provides additional numerical insights.

II. Model and Definitions

A. Preliminaries

We begin by defining a stage game.

Definition 1 (Stage Game): A two agent stage game $\mathcal{G}$ is a tuple $(A_1, A_2, \mathcal{R}_1, \mathcal{R}_2)$, where A_i denotes a finite set of actions available and $\mathcal{R}_i : A_1 \times A_2 \to \mathbb{R}^+$ is the payoff function, both for agent i, $i \in \{1, 2\}$. A mixed strategy π_i for agent i is a probability distribution over its action set A_i. A pure strategy is a strategy in which the probability of selecting a particular action is one.

Following standard notation, when referring to an agent i, we will refer to the other agent as $-i$. Stage games for two agents can be described using a bi-matrix whose rows (resp. columns) correspond to the possible actions of the first (resp. second) agent. The (j, k) entry of the bi-matrix is the pair of values $(r^1_{j,k}, r^2_{j,k})$, where $r^i_{j,k}$ denotes the payoffs to agent i when agent 1 plays action j and agent 2 plays action k. We use R^i to denote the payoff matrix of agent i that is obtained by collecting the entries $r^1_{j,k}$ and $r^2_{j,k}$, respectively.

Definition 2 (Nash Equilibrium): For a stage game, a strategy profile (π^*_i, π^*_{-i}) is a Nash equilibrium, and the pair $(\mathcal{R}_i(\pi^*_i, \pi^*_{-i}), \mathcal{R}_{-i}(\pi^*_i, \pi^*_{-i}))$ is a Nash outcome, if

$$\mathcal{R}_i\big(\pi^*_i, \pi^*_{-i}\big) \geq \mathcal{R}_i\big(\pi_i, \pi^*_{-i}\big), \forall \pi_i \neq \pi^*_i \quad \forall i \in \{1, 2\}.$$

A popular model for how agents can learn these equilibria is that of a *repeated game* in which the agents play a given stage game repeatedly. At each iteration or time step, the agent observes its (and possibly the other agent's) rewards and actions and updates its strategy based on its observations. One can categorize the observations of the agents as *perfect* or *imperfect monitoring*. With perfect monitoring, an agent can observe the actions selected and the payoffs obtained by all the agents. With imperfect monitoring, an agent observes the actions of all the agents but can observe only its own payoffs. The payoff matrix R^i is said to be *completely known* to agent i if agent i has the information of all entries of R^i. Similarly, a row j (resp. column k) of R^i is said to be *completely known* to agent i if agent i has the information of the all of the entries of jth row (resp. kth column) of R^i.

B. Fictitious Play and Regret-Matching

A *learning algorithm* for agent i is a mapping from the available observations to an action $a_i \in A_i$ at every iteration of the repeated game. Two classical learning algorithms are fictitious play and regret-matching, that are known to converge to the Nash and correlated equilibria, respectively, for a wide class of stage games [13], [14].

Fictitious Play: Let $\hat{\mathbf{a}}_{-i}(t)$ denote the vector of empirical frequencies of actions $a_{-i} \in A_{-i}$ played until time t. Then, in fictitious play, agent i selects action according to

$$a_i(t) = \arg\max_{a \in A_i} \mathcal{R}_i\big(a, \hat{\mathbf{a}}_{-i}(t-1)\big). \tag{1}$$

Regret-matching: Define the *instantaneous regret* of agent i at time t for action $a \in A_i$ as $\delta^t_i(a) := \mathcal{R}_i(a, a_{-i}(t)) - \mathcal{R}_i(a_i(t), a_{-i}(t))$ and define the average regret of agent i for action $a \in A_i$ at time T as $\delta^{\text{avg}}_{T,i}(a) := \frac{1}{T}\sum_{t=1}^{T} \delta^t_i(a)$. Let $\delta_+(a) := \max\{0, \delta^{\text{avg}}_{T,i}(a)\}$ and let $|\cdot|$ denote the cardinality of a set. Then, the regret-matching algorithm requires agent i to select action $a \in A_i$ with probabilities

$$p^i_t(a) = \begin{cases} \frac{\delta_+(a)}{\sum_{a' \in A_i} \delta_+(a')}, & \text{if } \sum_{a' \in A_i} \delta^{\text{avg}}_+(a') > 0, \\ \frac{1}{|A_i|}, & \text{otherwise.} \end{cases} \tag{2}$$

C. Model Considered

We consider an infinitely repeated stage game $\mathcal{G}$ under perfect monitoring where, at the first iteration, the agents do not have any information about their own and the other agent's payoff matrices. To make our framework applicable to existing learning algorithms, we follow a common model for the rewards in terms of their long term average as follows. Let $\mathcal{A}_i$ and $\mathcal{A}_{-i}$ denote the learning algorithm followed by agent i and agent $-i$, respectively. Given a stage game $\mathcal{G}$, the *value* for agent i is defined as $U_i(\mathcal{G}, \mathcal{A}_i, \mathcal{A}_{-i}) = \liminf_{T\to\infty} \mathbb{E}[\frac{1}{T}\sum_{t=0}^{T} \mathcal{R}_{i,t}]$, where $\mathcal{R}_{i,t}$ is the payoff received by agent i at time t. For notational ease, we drop the dependence on $\mathcal{G}$ and write the term as $U_i(\mathcal{A}_i, \mathcal{A}_{-i})$. Note that $U_i(\mathcal{A}_i, \mathcal{A}_{-i}) > 0$ since $\mathcal{R}_i > 0$. We say that agent i *deviates* from a prescribed learning algorithm $\mathcal{A}$ if its selects its actions according to any other algorithm $\mathcal{A}'$ in at least one interval of times $[t_1, t_2]$ for any $t_2 \geq t_1 \geq 0$. The following quantity characterizes how much an agent i gains by deviating from an algorithm $\mathcal{A}$.

Definition 3 (Rationality Ratio): Suppose both agents are prescribed an algorithm $\mathcal{A}$ and agent i deviates from $\mathcal{A}$ to algorithm $\mathcal{A}'$. Then, for any $i \in \{1, 2\}$, the *rationality ratio* of algorithm $\mathcal{A}$ is defined as

$$s\big(\mathcal{A}', \mathcal{A}\big) := \frac{U_i\big(\mathcal{A}', \mathcal{A}\big)}{U_i(\mathcal{A}, \mathcal{A})}. \tag{3}$$

Given a constant $c \geq 1$, the algorithm $\mathcal{A}$ is *c-rational* if

$$\sup_{\mathcal{G}, \mathcal{A}'} s\big(\mathcal{A}', \mathcal{A}\big) \leq c. \tag{4}$$

Finally, an algorithm $\mathcal{A}$ is *perfectly rational* if $c = 1$.

	Agent 2	
Agent 1	$(r^1_{1,1}, 2)$	(1,1)
	$(r^1_{2,1}, 1)$	**(5, 5)**

Fig. 1. A 2×2 game $\mathcal{G}$ for the proof of Theorem 1.

The value of using a multiplicative, rather than additive, measure to characterize incentives for deviation is that it is not sensitive to scale of the payoffs as opposed to typical additive measures, such as *game-theoretic regret* [15]. Further, c-rational algorithms provide a constant factor guarantee to the worst-case and thus, provides insights into whether (and by how much) the agents have an incentive to deviate from the algorithm in the worst-case.

Problem Statement. The aim of this letter is to determine whether the fictitious play and regret-matching algorithms are c-rational. If not, then to design and analyze c-rational algorithms with minimum c, especially in a manner that preserves convergence guarantees of such algorithms.

III. Irrationality of Existing Algorithms

In this section, we provide discouraging results; for fictitious play and regret-matching algorithms, a strategic agent has unbounded incentive to deviate from these algorithms.

Theorem 1: Regret-matching algorithm is not c-rational for any given constant $c \geq 1$.

Proof: Observe that the supremum in equation (4) is over $\mathcal{G}$ and any other algorithm $\mathcal{A}'$. We will construct a $\mathcal{G}$ and an algorithm $\mathcal{A}'$ for which equation (4) does not hold for any $c \geq 1$, even if the agents know the payoff matrices at the first iteration of the game $\mathcal{G}$. Without loss of generality, suppose agent 1 deviates from regret-matching algorithm. Consider a 2×2 bi-matrix game $\mathcal{G}$ as shown in Figure 1 with entry $r^1_{1,1} = 5(c+1)$ and $r^1_{2,1} > r^1_{1,1}$. It can be verified that when both agents follow algorithm $\mathcal{A}$, the strategies converge to the pure Nash equilibrium or the $(2, 2)$ entry. Thus, $U_1(\mathcal{A}, \mathcal{A}) = 5$. Next, consider Algorithm $\mathcal{A}'$ which, at each time t, has agent 1 play row $j = 1$. Suppose that at time t, agent 2 selects action $k = 1$. The instantaneous regret for agent 2 for not selecting column $k = 2$ is $\delta^t_2(2) = r^2_{1,2} - r^2_{1,1} = -1$. Similarly, if agent 2 selects action $k = 2$, then the instantaneous regret for not selecting column $k = 1$ is $\delta^t_2(1) = r^2_{1,1} - r^2_{1,2} = 1$. Since the action of agent 1 does not change at any time t, it follows that at every time t at which agent 2 selects column 2, agent 2 experiences a positive regret. Thus, as $t \to \infty$, from equation (2), $p^2_{t+1}(1) \to 1$. This implies $U_1(\mathcal{A}', \mathcal{A}) = r^1_{1,1}$. Since $r^1_{1,1} = 5(c+1)$, it follows that $s(\mathcal{A}', \mathcal{A}) = c + 1$ which implies $\sup_{\mathcal{G},\mathcal{A}'} s(\mathcal{A}', \mathcal{A}) \geq c+1$. This means that equation (4) can never hold for any given constant $c \geq 1$. ■

Theorem 2: Fictitious play algorithm is not c-rational for any given constant $c \geq 1$.

Proof: The proof is analogous to the proof of Theorem 1 and is omitted due to space constraints. ■

	Agent 2	
Agent 1	(5,8)	(1,9)
	(3,3)	**(2, 10)**

(a) Game $\mathcal{G}_1$.

	Agent 2	
Agent 1	**(5, 10)**	(1,9)
	(3,10(c + 2))	(2,10(c + 1))

(b) Game $\mathcal{G}_2$.

Fig. 2. Games $\mathcal{G}_1$ and $\mathcal{G}_2$ for the proof of Theorem 3. The equilibrium entries are highlighted in bold.

IV. Nonexistence of Rational Algorithms Under Imperfect Monitoring

We now establish that, under imperfect monitoring, no c-rational algorithms exist for certain classes of games.

Theorem 3: Under imperfect monitoring, there exist games for which no algorithm is c-rational for any $c \geq 1$.

Proof: Similar to the proof of Theorem 1, we will construct two stage games $\mathcal{G}_1$ and $\mathcal{G}_2$ and show that by restricting ourselves to only two stage games, no c-rational algorithms exist. The result would then follow given the supremum operator. Further, we will prove the result in a restrictive setting that the agents know that the game is either $\mathcal{G}_1$ or $\mathcal{G}_2$ and the payoff matrices associated with these games. As the original setting considered in this letter is a generalization, the result naturally will hold for the general setting as well.

Without loss of generality, suppose that agent 2 deviates from an algorithm $\mathcal{A}$ and follows algorithm $\mathcal{A}'$. Consider two stage games $\mathcal{G}_1$ and $\mathcal{G}_2$ as depicted in Figure 2. In game $\mathcal{G}_1$ (resp. $\mathcal{G}_2$), the entry $(2, 2)$ (resp. entry $(1, 1)$) is the only possible equilibrium because of domination. Suppose that, if both agents select their actions according to an algorithm $\mathcal{A}$, their respective strategies converge to the entries corresponding to the equilibrium of the game.

Suppose agent 2 always selects column 2 and the game is $\mathcal{G}_2$. Further, even by assuming that Algorithm $\mathcal{A}$ has the information that the game selected is either $\mathcal{G}_1$ and $\mathcal{G}_2$ and completely knows its own payoff matrices, $\mathcal{A}$ cannot determine the actual game being played by the agents. This is because of the imperfect monitoring setting and that the payoffs for agent 1 is identical in both $\mathcal{G}_1$ and $\mathcal{G}_2$. Thus, given that agent 1 can only observe agent 2's actions and since agent 2 selects only column 2, algorithm $\mathcal{A}$ selects row 2 for agent 1. This is because, given that agent 2 selects column 2, selecting row 1 yields a lower payoff for agent 1. This means that $U_2(\mathcal{A}, \mathcal{A}') = 10(c+1)$. Thus, even by restricting the set of games to only $\mathcal{G}_1$ and $\mathcal{G}_2$, it follows that $\sup_{\mathcal{G}_1,\mathcal{G}_2,\mathcal{A}'} s(\mathcal{A}, \mathcal{A}') = c+1$ for any constant c. This implies that $\sup_{\mathcal{G},\mathcal{A}'} s(\mathcal{A}, \mathcal{A}') = c+1$ and the result follows. ■

Given these negative results, it is natural to ask whether *any* c-rational learning algorithms exist. Fortunately, the answer is in the affirmative. In the next section, we present an algorithm that utilizes any existing learning algorithm (e.g., fictitious play or regret-matching) and is provably perfectly rational under mild assumptions.

V. Rational Learning Algorithms

In light of Theorem 3, we impose the following assumption for the rest of this letter.

Assumption 1 (Perfect Monitoring): Every agent perfectly observes the payoffs and the actions of the other agent at each time.

Note that there exists many classes of games, such as zero-sum games or common interest games, which implicitly have a perfect monitoring setting. Let $\mathcal{A}$ denote any existing learning algorithm. Our algorithm, which we call Algorithm Rational-$\mathcal{A}$, takes $\mathcal{A}$ as an input and specifies two strategies in a way akin to the grim trigger strategies [16]. The first is a strategy that is followed by agent i as long as agent $-i$ follows the prescribed algorithm. However, if agent $-i$ is detected to deviate, agent i switches to a prescribed *punishment* strategy. Thus, Algorithm Rational-$\mathcal{A}$ can consist of two phases; the self-play phase and the punishment phase. Since the agents do not have any information about the payoff matrices of the stage game $\mathcal{G}$, the self-play phase itself needs to consist of two sub-phases, as described below.

Exploration Sub-Phase: In the exploration sub-phase, every agent maintains and updates a local estimate of the payoff matrix of the other agent. To this end, the agents select their actions according to a joint sequence specified at time 0. Specifically, at time $1 \leq t \leq |A_1||A_2|$, agent 1 sequentially selects each row $1 \leq j \leq |A_1|$, starting with the first row, $|A_2|$ number of times. Simultaneously, agent 2 selects each column sequentially. Once agent 2 selects the last column, it repeats the sequence. This ensures that each entry of the payoff matrices is revealed sequentially over time. Further, since the exploration strategy for both agents is deterministic, any deviation is guaranteed to be detected.

Exploitation Sub-Phase: The agents enter this sub-phase once the exploration sub-phase ends and if no agent has deviated from the prescribed algorithm. In this sub-phase, the agents select their actions based on Algorithm $\mathcal{A}$ until an agent detects that the other agent has deviated from it.

If the strategy prescribed for the agents is deterministic, such as in the exploration sub-phase or due to some specific $\mathcal{A}$, the detection of whether an agent deviated or not is ensured due to Assumption 1. Thus, to describe how to detect whether an agent i has deviated or not, we assume that the strategy followed by the agent i is stochastic. In this case, agent $-i$ must compare the empirical probability distribution over the actions selected by agent i with the probability distribution over the actions that agent i should have selected its actions from. To achieve this, the exploitation sub-phase of Algorithm Rational-$\mathcal{A}$ runs in epochs, each consisting of a finite number of N_t iterations. As the payoff matrices are completely known in the exploitation sub-phase, given algorithm $\mathcal{A}$, agent i determines the probability distribution over the actions of agent $-i$ as well as agent i. Let ϕ_t^i and ϕ_t^{-i} denote the probability distribution over the actions from which agent i and agent $-i$, respectively, must choose their actions from, if they were following algorithm $\mathcal{A}$. In every iteration $n \leq N_t$ of epoch t, agent i selects action according to ϕ_t^i and observes the action selected by agent $-i$. Once the epoch ends, i.e., after N_t iterations, agent i computes the empirical cumulative distribution function (CDF), denoted as F_t^{-i}, from the observed actions of agent $-i$. Let $\mathcal{F}_t^{-i}(x)$ denote the CDF determined using ϕ_t^{-i}. Then, after computing the empirical CDF, agent i checks whether the following condition holds at the end of epoch t:

$$\sup_{x \in \mathbb{R}} |\mathcal{F}_t^{-i} - F_t^{-i}| > \epsilon_t, \tag{5}$$

where $\epsilon_t = \frac{1}{t}$. If condition (5) holds, agent i proceeds to the next epoch $t+1$. If not, agent i enters the punishment phase.

We now briefly comment on the choice of N_t and ϵ_t. If Algorithm $\mathcal{A}$ is such that it selects actions for an agent deterministically, then $N_t = 1$. Otherwise, $N_t = \frac{c_1 \log(\frac{c_2 t}{\gamma})}{\epsilon_t^2}$, where $\gamma \in (0,1)$ and $c_1 > 0$ and $c_2 > 1$ are some real numbers satisfying $2 \geq c_2 t^{2c_1 - 1}$. Although the choice of N_t will be clear from the proof of Theorem 4, we provide an intuition behind this choice. On one hand, we require that in case agent i does not deviate, then the equilibrium strategies of Algorithm Rational-$\mathcal{A}$ must converge to that of when the agents would have selected actions according to $\mathcal{A}$. To achieve this, we must ensure that equation (5) holds with very low probability (almost 0), when agent i does not deviate. On the other hand, in case agent i deviates, we require the algorithm to enter the punishment phase. Thus, motivated from [17], we select N_t (resp. ϵ_t) such that it increases (resp. decreases) in every epoch t.

Without loss of generality, we assume that agent 1 deviates and refer to it as the adversary. Further, we denote the local estimate of the payoff matrix of agent 1 that agent 2 maintains as $\hat{R}^1$. Note that, when the payoff matrix of agent 1 is completely known to agent 2, $\hat{R}^1 = R^1$.

Punishment phase: The idea is to *punish* the adversary for not adhering to the algorithm. Since the adversary can deviate from the algorithm either during the exploration or the exploitation sub-phase, the punishment strategy depends on when the adversary deviates. We begin with the definition of the *minimax* strategy which is used in the punishment phase.

Definition 4: The minimax value for agent 1 on some matrix Q is defined as $\bar{V}_1(Q) = \min_{y \in \mathcal{Y}} \max_{z \in \mathcal{Z}} y^\top Q z$, where $\mathcal{Y}$ (resp. $\mathcal{Z}$) denotes the set of all probability distributions over the pure strategy $|A_1|$ (resp. $|A_2|$) and the corresponding policy y^* is called the minimax strategy for agent 1. Analogous definition holds for agent 2.

Let t denote the time when the punishment phase begins. In the exploitation sub-phase, the payoff matrices $\hat{R}^1$ and R^2 are completely known by agent 2. Therefore, if the deviation is during the exploitation sub-phase, the punishment strategy is to select an action for agent 2 by computing the minimax strategy on matrix $\hat{R}^1$ and execute it for all time $\tau \geq t$.

Given the punishment strategy when an adversary deviates during the exploitation phase, an adversary might be tempted to deviate during the exploration phase. This is because, since the agents do not know the game payoff completely, computing the true minimax strategy is not possible. Consequently, it may be possible that the adversary may obtain a better payoff upon deviating from the exploration phase as opposed to deviating in the exploitation phase. To address this, we now describe the punishment strategy for when the adversary deviates during the exploration phase.

If agent 1 deviates during the exploration sub-phase, since the payoff matrix R^1 is not completely known to agent 2,

the minimax strategy on $\hat{R}^1$ cannot be computed. Agent 2 constructs a payoff matrix $\tilde{R}^1$ corresponding to the local estimate of the adversary's original payoff $\hat{R}^1$ such that the entries that are known in $\hat{R}^1$ are the same in $\tilde{R}^1$, while the entries that are not known in $\hat{R}^1$ are substituted as 0. Then, agent 2 selects an action with equal probability of $\frac{1}{|A_2|}$ until at least one of the rows of matrix $\hat{R}^1$ is completely known,[1] updating the unknown entries of $\tilde{R}^1$ as they are revealed. If agent 2 deviates instead of agent 1, then agent 1 selects an action with equal probability of $\frac{1}{|A_1|}$ until at least one of the columns of matrix $\hat{R}^2$ is completely known. Once at least one of the rows of $\hat{R}^1$ is completely known, agent 2 then computes and executes the minimax strategy on matrix $\tilde{R}^1$. If no new entry of matrix R^1 is revealed, agent 2 continues to play the computed minimax strategy. Otherwise, agent 2 updates $\hat{R}^1$ and $\tilde{R}^1$ and recomputes the minimax strategy on $\tilde{R}^1$.

Lemma 1: Let $\mathcal{A}$ denote an algorithm that incorporates punishment phase and suppose agent i deviates. Let $\bar{V}_i^p := \min_{a_{-i}\in A_{-i}} \max_{a_i\in A_i} r^i_{a_i,a_{-i}}$. Then, for a given $c \geq 1$, algorithm $\mathcal{A}$ is perfectly rational if $\bar{V}_i^p \leq cU_i(\mathcal{A},\mathcal{A})$ holds.

Proof: Without loss of generality, suppose that agent 1 deviates from algorithm $\mathcal{A}$. First, consider that algorithm $\mathcal{A}$ enters the punishment phase from the exploitation sub-phase. Then, agent 2 selects action according to the minimax strategy, defined in Definition 4, on matrix $\hat{R}^1 = R^1$. As $t \to \infty$, $U_1(\mathcal{A}',\mathcal{A}) \to \bar{V}_1(\hat{R}^1)$. Since $\bar{V}_1^p(\hat{R}^1) \geq \bar{V}_1(\hat{R}^1)$ [18] and given the condition in Lemma 1, we obtain

$$\frac{U_1(\mathcal{A}',\mathcal{A})}{U_1(\mathcal{A},\mathcal{A})} = \frac{\bar{V}_1(R^1)}{U_1(\mathcal{A},\mathcal{A})} \leq \frac{\bar{V}_1^p(R^1)}{U_1(\mathcal{A},\mathcal{A})} \leq c. \tag{6}$$

Equation (6) holds even in the case when algorithm $\mathcal{A}$ enters the punishment phase from the exploration sub-phase and there exists a time t at which the matrix R^1 is completely known by agent 2. In the case when none of the rows of the payoff matrix R^1 is completely known by agent 2, at any time t, there is a positive probability that a new entry of R^1 will be revealed. Thus, there exists a time t at which at least one of the row of matrix R^1 will be completely known by agent 2. Thus, in what follows, we consider the case for which the following jointly hold: (i) algorithm $\mathcal{A}$ enters the punishment phase from the exploration sub-phase, (ii) at least one of the rows of R^1 (say the j-th row) is completely known by agent 2, and (iii) the matrix R^1 is not completely known by agent 2 at any time t. Let τ denote the time when an entry of matrix R^1 was revealed for the last time. As $t \to \infty$ and since no new entry of R^1 is revealed after time τ, $U_1(\mathcal{A}',\mathcal{A}) \to \bar{V}_1(\tilde{R}^1)$. Since $\bar{V}_1^p(\tilde{R}^1) \geq \bar{V}_1(\tilde{R}^1)$ [18], we now show that $\bar{V}_1^p(\tilde{R}^1) \leq \bar{V}_1^p(R^1)$. Suppose that entry (j',k) for any $j' \neq j$ of matrix R^1 is not known by agent 2. Let $\tilde{R}^1_{j'}$ denote the matrix if the entry (j',k) was known by agent 2. Then, if $r^1_{j',k} > r^1_{j,k}$, it follows that $\bar{V}_1^p(\tilde{R}^1_{j'}) > \bar{V}_1^p(\tilde{R}^1)$. This is because the entry $r^1_{j',k} = 0$ in matrix $\tilde{R}^1$. Further, if $r^1_{j',k} \leq r^1_{j,k}$, it follows that $\bar{V}_1^p(\tilde{R}^1_{j'}) = \bar{V}_1^p(\tilde{R}^1)$. Thus, for any (j',k) entry that is not known by agent 2, $\bar{V}_1^p(\tilde{R}^1_{j'}) \geq \bar{V}_1^p(\tilde{R}^1)$ which implies that $\bar{V}_1^p(R^1) \geq \bar{V}_1^p(\tilde{R}^1)$ and the result follows. ■

[1]The agent skips this step if any one row of $\hat{R}^1$ is known to begin with.

Remark 1: In this letter, we consider agents to be rational, i.e., agent that always pursue its best interest, as opposed to an adversary. The punishment strategy acts only as a threat for the automated agents as it ensures that there is no incentive for an agent to deviate from its prescribed learning algorithm. This implies that, if there is no incentive for an agent to deviate from an algorithm, the agents do not enter the punishment phase. Since we consider automated learning agents in this letter, such strategies provide credible threats. However, such strategies may not be credible against an adversarial agent that may not aim to maximize its payoff.

Theorem 4: Let $\mathcal{A}$ be any existing learning algorithm for repeated games and let π_1^* and π_2^* denote the strategies of agent 1 and agent 2, respectively, if they selected actions according to $\mathcal{A}$. Then, for a given $\gamma \in (0,1)$ and a given constant $c \geq 1$, Algorithm Rational-$\mathcal{A}$ is

1) c-rational if Lemma 1 holds.
2) If π_1^* and π_2^* converge to an equilibrium then, with probability at least $1-\gamma$, so does the strategies of the agents when they follow Algorithm Rational-$\mathcal{A}$.

Proof: Without loss of generality, suppose that agent 1 deviates from Algorithm Rational-$\mathcal{A}$. If agent 1 deviates during exploration sub-phase, then any deviation is guaranteed to be detected and Algorithm Rational-$\mathcal{A}$ is sure to enter the punishment phase. Thus, Lemma 1 yields that, in this case, Algorithm Rational-$\mathcal{A}$ is perfectly rational. We now consider the case when an agent deviates during the exploitation sub-phase and that $\mathcal{A}$ represents an algorithm that selects an action based on a probability distribution. Observe that in any epoch t and in the worst-case, agent 1 can select its action such that at after N_t iterations, the condition $\sup_{x\in\mathbb{R}} |\mathcal{F}_t^1(x) - F_t^1(x)| > \epsilon_t$ does not hold. Further, for any epoch t, as $\epsilon_t = \frac{1}{t}$, it follows that as $t \to \infty$, $\epsilon_t \to 0$. Thus, for a high value of t, there are two cases.

Case 1: Suppose after some epoch t, since $\epsilon_t \approx 0$, the condition defined in equation (5) holds. Then, the algorithm enters punishment phase and (1) holds from Lemma 1.

Case 2: The second case is that, since $\epsilon_t \to 0$ as $t \to \infty$, agent 1 starts selecting actions according to Algorithm $\mathcal{A}$ and does not deviate. This means that from this moment on, both agents select their actions according to Algorithm $\mathcal{A}$. For any epoch t, using [19, Th. 1], the probability that condition defined in equation (5) holds is at most $2e^{-2N_t\epsilon_t^2}$. By taking the union bound over all t, the probability that the condition in equation (5) never holds is at least $1 - t2e^{-2N_t\epsilon_t^2}$, which reduces to at least $1 - \frac{2\gamma}{c_2 t^{2c_1-1}}$ given the choice of N_t. By selecting c_1 and c_2 such that $\frac{2\gamma}{c_2 t^{2c_1-1}} \geq \gamma$, with probability at least $1-\gamma$, the condition in equation (5) never holds until epoch t and Algorithm Rational-$\mathcal{A}$ does not enter the punishment phase until epoch t. Hence, if π_1^* and π_2^* converge to an equilibrium, then so do the strategies obtained from Rational-$\mathcal{A}$. The proof when no agent deviates and the strategies obtained from Algorithm Rational-$\mathcal{A}$ converges to the equilibrium, with probability $1-\gamma$, if π_1^* and π_2^* converge to an equilibrium is analogous to that of Case 2. Finally, the proof when $\mathcal{A}$ represents an algorithm that selects an action determinisitically is similar to that when

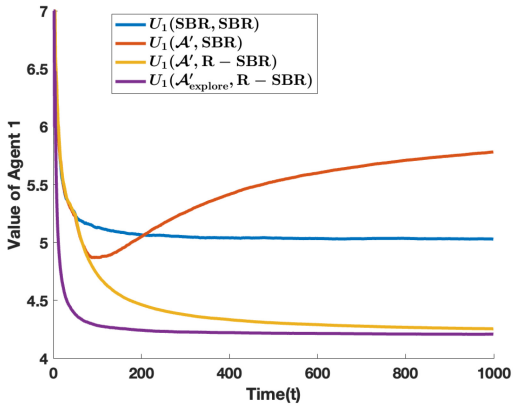

Fig. 3. Value of agent 1 (adversary) over time. Since $U_1(\text{SBR}, \text{SBR}) = U_1(\text{R-SBR}, \text{R-SBR})$, the curve for $U_1(\text{SBR}, \text{SBR})$ is not shown.

the agents deviate during exploration phase and has been omitted. ■

Algorithm Rational-$\mathcal{A}$ not only incorporate algorithms that are known to converge to an equilibrium, but also algorithms for which convergence results are not currently known.

Remark 2: Our work easily extends to the case of n agents if a central entity can inform the agents about the deviation. In this case, all the remaining agents can jointly punish the agents who have deviated. Our work also extends to stochastic games under the assumption that the current state and the transition probabilities is known, the agents follow stationary strategies and the underlying game is irreducible.

VI. Numerical Results

For the numerical results, we consider that agents are prescribed smoothed best-response (SBR) algorithm [3] and Rational-SBR algorithm with smoothing parameter set to 1. We consider the game analogous to that in Figure 1. It can be checked that for this game, the condition described in Lemma 1 holds with $c = 1$. We provide additional numerical results in [20]. Note that when agent i deviates during the exploration (resp. exploitation) sub-phase, then the algorithm followed by agent i is denoted as $\mathcal{A}'_{\text{explore}}$ (resp. $\mathcal{A}'$). Further, as the agents follow SBR and since Rational-SBR detects a deviation with high probability, all of our numerical results represent the mean over 50 runs.

Figure 3 illustrates the value of agent 1 when (i) both agents follow Rational-SBR (R-SBR), (ii) agent 1 deviates and agent 2 follows SBR, (iii) agent 2 follows R-SBR and agent 1 deviates in the exploitation phase, and (iv) agent 2 follows R-SBR and agent 1 deviates in the exploration phase. The time at which the adversary deviates was set to $t = 3$ and $t = 50$ for the exploration and the exploitation sub-phase, respectively. We consider that agent 1 deviates to the strategy described in [4] which may be sub-optimal against SBR algorithm.

Figure 3 illustrates that even deviating to a sub-optimal strategy, agent 1 achieves a higher payoff. This means that there exist games for which Algorithm SBR may not be c-rational, for $c \geq 1$. Further, from Figure 3, $U_1(\mathcal{A}', \text{R-SBR}) < U_1(\text{R-SBR}, \text{R-SBR})$ and $U_1(\mathcal{A}'_{\text{explore}}, \text{R-SBR}) < U_1(\text{R-SBR}, \text{R-SBR})$. This implies that Algorithm R-SBR is perfectly rational (Theorem 4).

VII. Conclusion

We considered a two-agent non-cooperative repeated game framework and defined the rationality ratio as the most an agent can obtain by deviating from a learning algorithm to their payoff from following it. A learning algorithm is called c-rational if its rationality ratio is at most c. We first established that fictitious play and regret-matching algorithm are not c-rational for any given constant c. We also established that there exist classes of games in which a c-rational algorithm does not exist under imperfect monitoring. We then presented an algorithm that is provably c-rational for a given $c \geq 1$ under mild assumptions.

References

[1] Y. Yang and J. Wang, "An overview of multi-agent reinforcement learning from game theoretical perspective," 2020, *arXiv:2011.00583*.
[2] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of Reinforcement Learning and Control*. Cham, Switzerland: Springer, 2021, pp. 321–384.
[3] D. Fudenberg and D. K. Levine, *The Theory of Learning in Games*, vol. 2. Cambridge, MA, USA: MIT Press, 1998.
[4] B. Vundurthy, A. Kanellopoulos, V. Gupta, and K. G. Vamvoudakis, "Intelligent players in a fictitious play framework," *IEEE Trans. Autom. Control*, vol. 69, no. 1, pp. 479–486, Jan. 2024.
[5] Y. Arslantas, E. Yuceel, and M. O. Sayin, "Strategizing against Q-learners: A control-theoretical approach," 2024, *arXiv:2403.08906*.
[6] Y. Kolumbus and N. Nisan, "How and why to manipulate your own agent: On the incentives of users of learning agents," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 28080–28094.
[7] R. Brafman and M. Tennenholtz, "Efficient learning equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 15, 2002, pp. 1–8.
[8] A. DiGiovanni and A. Tewari, "Balancing adaptability and non-exploitability in repeated games," in *Proc. Uncertainty Artif. Intell.*, 2022, pp. 559–568.
[9] M. Braverman, J. Mao, J. Schneider, and M. Weinberg, "Selling to a no-regret buyer," in *Proc. ACM Conf. Econ. Comput.*, 2018, pp. 523–538.
[10] Y. Deng, J. Schneider, and B. Sivan, "Strategizing against no-regret learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.
[11] E. Kalai and E. Lehrer, "Rational learning leads to Nash equilibrium," *Econometrica*, vol. 61, no. 5, pp. 1019–1045, 1993.
[12] D. P. Foster and H. P. Young, "On the impossibility of predicting the behavior of rational agents," *Proc. Nat. Acad. Sci.*, vol. 98, no. 22, pp. 12848–12853, 2001.
[13] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, 2000.
[14] G. W. Brown, "Iterative solution of games by fictitious play," *Act. Anal. Prod Allocat.*, vol. 13, no. 1, p. 374, 1951.
[15] Y. Vorobeychik and M. P. Wellman, "Stochastic search methods for Nash equilibrium approximation in simulation-based games," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2008, pp. 1055–1062.
[16] R. J. Aumann and L. S. Shapley, "Long-term competition—A game-theoretic analysis," in *Essays in Game Theory: In Honor of Michael Maschler*. New York, NY, USA: Springer, 1994, pp. 1–15.
[17] V. Conitzer and T. Sandholm, "AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents," *Mach. Learn.*, vol. 67, pp. 23–43, May 2007.
[18] J. P. Hespanha, *Noncooperative Game Theory: An Introduction for Engineers and Computer Scientists*. Princeton, NJ, USA: Princeton Univ. Press, 2017.
[19] P. Massart, "The tight constant in the Dvoretzky–Kiefer–Wolfowitz inequality," *Ann. Probab.*, vol. 18, no. 3, pp. 1269–1283, 1990.
[20] S. Bajaj, P. Das, Y. Vorobeychik, and V. Gupta, "Rationality of learning algorithms in repeated normal-form games," 2024, *arXiv:2402.08747*.