Last Round Convergence and No-Dynamic Regret in Asymmetric Repeated Games

Le Cong Dinh L.C.DINH@SOTON.AC.UK
School of Electronics and Computer Science, University of Southampton, United Kingdom

Tri-Dung Nguyen T.D.NGUYEN@SOTON.AC.UK
Alain B. Zemkoho A.B.ZEMKOHO@SOTON.AC.UK
School of Mathematical Sciences & CORMSIS, University of Southampton, United Kingdom

Long Tran-Thanh LONG.TRAN-THANH@WARWICK.AC.UK
Department of Computer Science, University of Warwick, United Kingdom

Abstract

This paper considers repeated games in which one player has a different objective than others. In particular, we investigate repeated two-player zero-sum games where the column player not only aims to minimize her regret but also stabilize the actions. Suppose that while repeatedly playing this game, the row player chooses her strategy at each round by using a no-regret algorithm to minimize her regret. We develop a no-dynamic regret algorithm for the column player to exhibit last round convergence to a minimax equilibrium. We show that our algorithm is efficient against a large set of popular no-regret algorithms the row player can use, including the multiplicative weights update algorithm, general follow-the-regularized-leader and any no-regret algorithms satisfy a property so called "stability".

Keywords: last round convergence, no-dynamic regret, asymmetric game, zero-sum game

1. Introduction

Repeated two-player zero-sum games form one of the most studied classes of repeated games in game theory. In this setting, thanks to Blackwell's famous approachability theorem, if a player's strategies are generated by algorithms (i.e., policies) with a special property called "no-regret", one can prove that, on average, that player does not perform worse than the best-fixed strategy in hindsight. A direct implication of this result is that if both players choose to play such no-regret algorithms, their average payoffs will converge to the game's minimax value. Put differently, the players' strategies will converge to a minimax equilibrium on average (e.g., see Cesa-Bianchi and Lugosi (2006) or Arora et al. (2012) for more details). It can also be easily shown that this (on-average) convergence holds independently from the prior information that each player has about the payoff matrix $\boldsymbol{A}$. That is, no matter how much prior information a player has about the game, she cannot exploit the other player's average payoff if the latter uses a no-regret algorithm.

This paper considers a shift of interest for the column player and investigates whether she can achieve that in the repeated two-player zero-sum games setting. Along with optimising the usual performance measure (i.e., no-regret property), the column player also wants to keep her strategy stable while repeatedly play the game. This is motivated by the fact that changing strategies through repeated games might be undesirable. For example, changing the (mixed) strategy of a company will increase the cost of operation to implement the new mixed strategy (e.g., as a result of having to

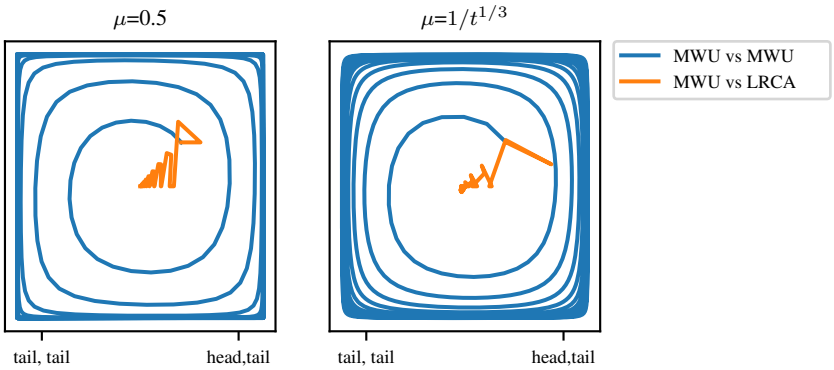

Figure 1: Player Strategies Spiraling Outwards In MWU vs Last Round Convergence in LRCA in Matching Pennies after 2500 iterations with the Same Initial Condition.

hire new equipment and employees). Therefore, the company often aims to maximise the revenue (i.e., the average payoff) and reduce the cost of operation by having a stable strategy. For another example, consider a government-owned company, for whom, along with the average benefit, keeping the market stable is one of the key goals to increase social welfare. Finally, in system design, the designer (the column player) will want the participant (the row player) to play a particular strategy so that the system is well behaved. In the online learning literature, maximising the average payoff and achieving the system's stability are often viewed as conflicting goals. That is, if all the player in a system follows a selfish behaviour (e.g., an FTRL no-regret algorithm) to maximise their payoff, then the dynamic of the system could become chaotic, and last round convergence never happens (e.g., see Mertikopoulos et al. (2018) for more details). Figure 1 demonstrates a simple game of matching pennies where the dynamic of two selfish players using Multiplicative Weight Update (MWU) (i.e., the blue trajectory) leads to outwards spirals with different step size μ. Thus, the Nash equilibrium point(i.e., the centre point) can never be achieved in this situation. The question is, whether there is a way to achieve both no-regret and stability in a system?

In this paper, we show that it is possible to exploit the no-regret algorithms to achieve both stability and no-dynamic regret; that is, the regret compared to each round's best action. In the fully adversarial setting, it is impossible to achieve no-dynamic regret property and therefore, state-of-the-art algorithms instead measure the success by comparing the regret with the best-fixed strategy in hindsight. By looking deeper into the adversarial setting and analysing strategic players' behaviour, one can achieve a much stronger concept of regret: dynamic regret.

The intuition behind this result can be explained as follows: If the row player believes that the goal for both players is to maximise their average payoffs(i.e., a fully adversary setting since the column player tries minimise the payoff of the row player in zero-sum game), then she will typically choose to play a no-regret "type" algorithm to achieve good average performance. Being aware of this, the column player can now choose an algorithm that exploits this information to have no-dynamic regret and last round convergence. We should note here, however, that it is not trivial how this can be efficiently done. For example, if the column player keeps playing the same strategy

(i.e., the minimax equilibrium), then while the system might achieve stability as the strategy of the row player will converge to the best response, this is not a no-regret algorithm and therefore, far away from being a no-dynamic regret algorithm.

1.1. Our contributions

Motivated by the abovementioned challenge, we propose a new algorithm that achieves no-dynamic regret for the column player in the case the row player follows a no-regret "type" algorithm. In contrast to normal no-regret algorithms that take best-fixed strategy as the milestone, dynamic regret(e.g., see Besbes et al. (2015)) compares the regret with the optimal strategy in hindsight. Thus, dynamic regret is a much stronger concept than normal regret, especially when every fixed strategy performs poorly in the game. In adversarial setting, we show that one player can leverage the other strategic player's behaviour to achieve a no-dynamic regret algorithm. In the general case, we introduce a method for the column player to have no-regret property against the row player's random strategies while still maintaining no-dynamic regret property against no-regret algorithm of the row player.

Furthermore, while on-average convergence has been extensively studied, it is still an open question whether last round convergence can be achieved, especially when the row player is also playing a no-regret algorithm (see Section 1.2 for more details). Against this background, we show that our algorithm, called the *L*ast *R*ound *C*onvergence in *A*symmetric games (LRCA), provably achieves last round convergence to a minimax equilibrium of the corresponding game. As shown in Figure 1, our Last Round Convergence Algorithm (LRCA) (i.e., the orange trajectory) converges to the Nash equilibrium of the Matching Pennies game while playing against the MWU with different step size μ. We prove that in our setting if the column player follows LRCA and the row player follows an algorithm from a wide set of common no-regret algorithms, then last round convergence to the minimax equilibrium of the game can be achieved. Note that in the case the horizon of play is unknown, the row player needs to employ a decreasing learning rate to make the algorithm no-regret. It means that the new observation feedback will be discounted compared to the old feedback. Lin et al. (2020) argues that this discounted new feedback is counter-intuitive and unjustifiable from economic principles. Thus, it is important to consider the no-regret learning dynamic where the learner does not impose a decreasing step size. In this paper, we allow the row player to play different algorithms, including μ-regret algorithms (i.e., constant step size) in which even the average convergence in self-play is not yet known.

Overall this paper has two main contributions. First, by allowing different strategy between the column and row player, we propose an algorithm that leads to last round convergence in many situations, which were proved not to hold (i.e., there is no last round convergence) in symmetric information settings; see Sections 3 for more details. Second, we show that by using the algorithm, the column player can achieve no-dynamic regret property; see Section 4 for more details. This answer the question of how to achieve both maximizing the average payoff and stability in a repeated game.

1.2. Related work

It is well-known that if both players use no-regret algorithms, their average strategies converge to a minimax equilibrium with the convergence rate of $\mathcal{O}(T^{-1/2})$; cf. Freund and Schapire (1999). Daskalakis et al. (2011) and Rakhlin and Sridharan (2013) have further improved this result by de-

veloping no-regret algorithms with near-optimal convergence rate of $\mathcal{O}(\frac{\log(T)}{T})$. However, despite the extensive literature on no-regret algorithms, these algorithms typically provide on-average convergence only, but not last round convergence. For example, Bailey and Piliouras (2018) proved that in games with an interior Nash equilibrium point, if the players use the multiplicative weights update (MWU) algorithm, then the last round strategy converges to the boundary. In addition, Mertikopoulos et al. (2018) showed that by using regularized learning, the system's behaviour is Poincare recurrent; that is, there is a loop in the strategy dynamics of the players. This undesirable feature causes many issues in game theory and applications, including unwanted cyclic behaviour in training Generative Adversarial Networks (GANs). Thus, a learning dynamic leading to last round convergence is of importance in the development of the field (e.g., see Daskalakis et al. (2017) for more details). Note that in a recent paper, Daskalakis and Panageas (2018) proved that if both players use the optimistic multiplicative weights update algorithm (OMWU), then we have last round convergence to the minimax equilibrium if this equilibrium point is unique. This last round convergence result also requires another restrictive assumption, namely: The constant step size of the update mechanism has to be calculated from the payoff matrix $\boldsymbol{A}$ of the game. Therefore, if the row player does not know the matrix $\boldsymbol{A}$ of the game, OMWU cannot guarantee the last round convergence (as it requires both players to know matrix $\boldsymbol{A}$). Besides, if the row player plays different no-regret algorithms such as MWU or FTRL, which are widely used in many applications, OMWU cannot lead to the last round convergence. This raises the question of whether there could be a robust algorithm, when playing against different no-regret algorithms, converging at the last round to minimax equilibrium.

1.3. Key assumptions

To proceed with the development of this paper, we make the following two assumptions:

1. The column player can get an arbitrarily close estimation of her minimax equilibrium.
2. The row player follows a no-regret "type" algorithm.

The rationale of these assumptions can be explained as follows: Assumption 1 can arise from asymmetric information two-player zero-sum games in which the column player knows the matrix $\boldsymbol{A}$ of the game. In this case, the column player can calculate the exact minimax equilibrium using linear programming. Realistic examples for this setting include problems from the security games domain, where an attacker can store the feedback from past observations and analyze the system's behaviour. Thus, the attacker could know the matrix $\boldsymbol{A}$ of the game. Another example comes from the perspective of a new company who enters an existing business market. In this market, every strategy and payoff of the players are revealed. Therefore, when a new company enters the market, they can anticipate their payoff for a particular action of their strategies. Thus, the new incomer knows the matrix $\boldsymbol{A}$ of the game. Note that the asymmetric game assumption might appear in many other applications, and hence we argue that this setting deserves attention from the online learning research community.

In symmetric information games (i.e., both players have the same prior information about the game), if the row player follows a no-regret type algorithm, the column player can first use a no-regret algorithm to estimate the minimax equilibrium. Note that in this estimation phase, the column player cannot guarantee the no-dynamic regret property like the asymmetric game. See Section 3.4 for more details.

Assumption 2 comes from the vanilla property of no-regret algorithms: without prior information, a player will not do worse than the best-fixed strategy in hindsight by following a no-regret algorithm. In this study, we allow the row player to deviate from a no-regret algorithm in a certain way; that is, she can choose a fixed learning rate. We also consider the full information feedback (see, e.g., Bailey and Piliouras (2018), Daskalakis et al. (2011), Freund and Schapire (1999)).[1]

Note that our setting differs from Daskalakis and Panageas (2018) in the following ways: we require neither the knowledge of the update step size nor the uniqueness of the minimax equilibrium. In addition, our result does not require the row player to follow the fixed learning rate OMWU. As such, we argue that our result can be applied to more real-world applications, due to its more reasonable and realistic assumptions (see Section 1.3 for more detailed discussions).

2. Preliminaries

Consider a repeated two-player zero-sum game. This game is described by a $n \times m$ payoff matrix $\boldsymbol{A}$ and w.l.o.g we assume the entries of $\boldsymbol{A}$ in $[0, 1]$. The rows and columns of $\boldsymbol{A}$ represent the *pure* strategies of the *row* and *column* players, respectively. We define the set of feasible strategies of the row player, at round t, by $\Delta_n := \{\boldsymbol{x}_t \in \mathbb{R}^n \mid \sum_{i=1}^n \boldsymbol{x}_t(i) = 1,\ \boldsymbol{x}_t(i) \geq 0\ \forall i \in \{1, ..., n\}\}$. The set of feasible strategies of the column player, denoted by Δ_m, is defined in a similar way. At round t, if the row (resp. column) player chooses a mixed strategy $\boldsymbol{x}_t \in \Delta_n$ (resp. $\boldsymbol{y}_t \in \Delta_m$), then the row player's payoff is $-\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$, while the column player's payoff is $\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$. Thus, the row (resp. column) player aims to minimise (resp. maximise) the quantity $\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$ (resp. $\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$). John von Neumann's minimax theorem Neumann (1928), founding stone in zero-sum games states that

$$\max_{\boldsymbol{y}\in\Delta_m} \min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \quad = \quad \min_{\boldsymbol{x}\in\Delta_n} \max_{\boldsymbol{y}\in\Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \quad = \quad v, \tag{1}$$

for some $v \in \mathbb{R}$. We call a point $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ satisfying the minimax theorem equation 1 *the minimax equilibrium of the game*. Throughout this paper, we use the notation $f(\boldsymbol{x}) := \max_{\boldsymbol{y}\in\Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}$. Since $\boldsymbol{A}$ is a non-zero matrix with entries in $[0, 1]$, we have $f(\boldsymbol{x}) \geq 0$. Note that $(\boldsymbol{x}_l, \boldsymbol{y}^*)$ which satisfy $f(\boldsymbol{x}_l) - v \leq \epsilon$ are ϵ-Nash equilibria(i.e., $\max_{\boldsymbol{y}\in\Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} - \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \leq \epsilon$ and $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} - \min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \leq \epsilon$) and $\epsilon = 0$ implies $\boldsymbol{x}_l$ is the Nash equilibrium of the row player. Similarly, if $\min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} = v$, then $\boldsymbol{y}$ is also a minimax equilibrium strategy. Next, we define the concept of a *no-dynamic regret* that will play an important role in this paper.

Definition 1 (Besbes et al. (2015)) *Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ be a sequence of mixed strategies played by the row player. An algorithm of the column player that generates a sequence of mixed strategies $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ is called a* no-dynamic regret *algorithm if we have*

$$\lim_{T\to\infty} \frac{DR_T}{T} = 0, \quad \textit{where} \quad DR_T := \sum_{t=1}^{\top} \left(\max_{\boldsymbol{y}\in\Delta_m} \boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y} - \boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t \right).$$

Here, the no-dynamic regret property is a stronger notion, compared to the usual no-regret property, as the latter, defined by $R_T = \max_{\boldsymbol{y}\in\Delta_m} \frac{1}{T} \sum_{t=1}^{\top} \boldsymbol{x}_t{}^\top \boldsymbol{A}(\boldsymbol{y} - \boldsymbol{y}_t)$, is benchmarked against the best fixed strategy in hindsight. Note that no-dynamic regret is typically impossible to achieve

1. Note that the main focus of this paper is on the investigation of the benefit of having asymmetric information. Thus, the analysis of other feedback cases, such as bandit or semi-bandit, is out of scope and remains part of future work.

with current state of the art algorithms in the adversarial symmetric setting. We will show that in our setting we can design an algorithm that can achieve the no-dynamic regret property.

Finally, it is important to mention that in this paper, we will use the Kullback-Leibler divergence to understand the behaviour of the row player's strategies.

Definition 2 (Kullback and Leibler (1951)) *The relative entropy or K-L divergence between two vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in Δ_n is defined as* $RE(\boldsymbol{x}_1\|\boldsymbol{x}_2) = \sum_{i=1}^{n} \boldsymbol{x}_1(i) \log\left(\frac{\boldsymbol{x}_1(i)}{\boldsymbol{x}_2(i)}\right)$.

The Kullback-Leibler divergence is always non-negative. Furthermore, from Gibbs's inequality (Mitrinovic and Vasic, 1970) we can show that $RE(\boldsymbol{x}_1\|\boldsymbol{x}_2) = 0$ if and only if $\boldsymbol{x}_1 = \boldsymbol{x}_2$ almost everywhere.

3. Last Round Convergence to Minimax Equilibrium

We first start with the investigation of last round convergence in asymmetric information cases. In particular, we present the Last Round Convergence of Asymmetric games (LRCA) algorithm for the column player. We then show that our algorithm is robust to many no-regret algorithms that can be played by the row player, namely: "*stable*" no-regret algorithms MWU/LMWU, general FTRL (i.e., it provides last round convergence when played against these algorithms). Under Assumption 1, we first study the case where the column player knows the exact minimax equilibrium $\boldsymbol{y}^*$ and the value v of the game (i.e. the column player knows the matrix $\boldsymbol{A}$ of the game). We then consider the case where only estimation of $\boldsymbol{y}^*$ and v are available to the column player in Section 3.4.

For a sequence of strategies $\boldsymbol{x}_1, \boldsymbol{x}_2, ...$ played by the row player, the LRCA algorithm (see Algorithm 1) for the column player can be described as follows: At each odd round, the column player plays the minimax equilibrium strategy, $\boldsymbol{y}^*$, so that in the next round, she cannot only predict the distance between the current strategy of the row player and a minimax equilibrium, but also prevent the row player from deviating the current strategy. Then, at the following even round, the column player chooses a strategy such that the feedback to the row player, $\boldsymbol{A}\boldsymbol{y}_t$, is a direction towards a minimax equilibrium strategy of the row player. Depending on the distance between the current strategy of the row player and a minimax equilibrium (which is measured by $f(\boldsymbol{x}_{t-1}) - v$), the column player chooses a suitable step size α_t so that the strategy of the row player will approach a minimax equilibrium. Note here that β is a constant number and we can fix $\beta = n^2$ so that our LRCA algorithm is robust against different no-regret algorithms that we consider in this paper. In order to obtain tighter regret convergence rate, we choose two different β in the case of MWU/LMWU and FTRL.

Algorithm 1 (LRCA) will work for a large set of learning rate, including the constant learning rate case. Simpler algorithms, such that "fictitious play" or "best response to the last feedback" will fail to converge in the simple case of constant learning rate and do not have the no-dynamic regret property in Section 4.

In Algorithm 1, every odd round the column player keeps playing the same strategy $\boldsymbol{y}^*$ and thus the row player can realize and exploit this pattern. To avoid this scenario, the column player can randomly choose two successive strategies such that: $\boldsymbol{y}_{2k-1} + \boldsymbol{y}_{2k} = \boldsymbol{y}^* + (1 - \alpha_{2k})\boldsymbol{y}^* + \alpha_{2k}\boldsymbol{e}_{2k}$ where α_{2k} and $\boldsymbol{e}_{2k}$ are chosen according to Algorithm 1. By following this method, the cumulative feedback received by the row player in the odd round will stay the same and thus the analysis of Algorithm 1 remains correct. We will prove in the following subsections that if the column player

follows LRCA and the row player uses one of the aforementioned no-regret algorithms, we will achieve last round convergence to the minimax equilibrium.

Algorithm 1: *L*ast *R*ound *C*onvergence in *A*symmetric algorithm (LRCA)

Input: Current iteration t, past feedback $\boldsymbol{x}_{t-1}^\top \boldsymbol{A}$ of the row player
Output: Strategy $\boldsymbol{y}_t$ for the column player
if $t = 2k-1,\ k \in \mathbb{N}$ **then**
  $\boldsymbol{y}_t = \boldsymbol{y}^*$
end
if $t = 2k,\ k \in \mathbb{N}$ **then**
  $\boldsymbol{e}_t := \operatorname{argmax}_{e \in \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots \boldsymbol{e}_m\}} \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}; \quad f(\boldsymbol{x}_{t-1}) := \max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}$
  $\alpha_t := \frac{f(\boldsymbol{x}_{t-1}) - v}{\beta}$
  $\boldsymbol{y}_t := (1-\alpha_t)\boldsymbol{y}^* + \alpha_t \boldsymbol{e}_t$
end

3.1. No-Regret Algorithms with Stability Property

We first prove that LRCA can work with a general class of no-regret algorithms, should the no-regret algorithm of the row player have a "stability" property defined below:

Definition 3 *A no-regret algorithm is* stable *if* $\forall t : \boldsymbol{y}_t = \boldsymbol{y}^* \implies \boldsymbol{x}_{t+1} = \boldsymbol{x}_t$.

The stability property is natural and can be explained as follows. Intuitively, in a good situation where the column player follows $\boldsymbol{y}^*$ and all the rewards $\boldsymbol{A}\boldsymbol{y}^*$ are equal (i.e., $\boldsymbol{A}\boldsymbol{y}^* = [v, v, \ldots, v]$), the row player's all strategy will provide the same reward and there is no incentive to change the current strategy. Thus, the next strategy of the row player will be the same as the current strategy (i.e., $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$) and the column player can use this information to exploit the row player in the next iteration.

There are many no-regret algorithms with stability property. For example, we prove below that a wide class of no-regret algorithms, Follow The Regularized Leader (FTRL), are stable:

Lemma 4 *Suppose the row player follows a FTRL algorithm with regularizer $R(\boldsymbol{x})$ defined as:*

$$\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x} \in \Delta_n} \mu \boldsymbol{x}^\top \left(\sum_{i=1}^{t-1} \boldsymbol{A}\boldsymbol{y}_i \right) + R(\boldsymbol{x}).$$

If there exists a fully-mixed minimax equilibrium strategy for the row player, then FTRL is stable.

Proof As there exists a fully-mixed minimax equilibrium strategy for the row player, we have $\boldsymbol{A}\boldsymbol{y}^* = [v, \ldots, v]^T$. Thus, we have:

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^* = v \quad \forall \boldsymbol{x} \in \Delta_n.$$

When the column player follows the minimax strategy, the minimization for $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ only differ by a constant term v, so their solutions are the same. ■

Note that the FTRL framework, with appropriately chosen $R(\boldsymbol{x})$, can recover many popular no-regret algorithms, such as online mirror descent, multiplicative weights update, and Hannan's algorithm (a.k.a. Follow the Perturbed Leader). See, e.g., (McMahan, 2011; Arora et al., 2012) for more details. Following the stability property, we prove the LRCA can lead to ϵ-Nash equilibrium, where ϵ can be chosen arbitrarily small:

Theorem 5 *Assume that the row player follows a stable no-regret algorithm and n is the row player's strategy dimension. Then, by following LRCA, for any $\epsilon > 0$, there exists $l \in \mathbb{N}$ such that $\frac{\mathcal{R}_l}{l} = \mathcal{O}(\frac{\epsilon^2}{n})$ and $f(\boldsymbol{x}_l) - v \leq \epsilon$.*

The proof is given in Appendix A. For no-regret algorithms with optimal regret bound $\mathcal{R}_l = O(\sqrt{l})$, following Theorem 5, the players will reach an ϵ-Nash equilibrium in at most $\mathcal{O}(\frac{1}{\epsilon})^4$ rounds. Due to the full information feedback assumption, the column player will know when the row player plays an ϵ-Nash equilibrium strategy. Depending on the number of rounds, the column player can lead the row player to play any ϵ-Nash equilibrium, after that switching from LRCA to $\boldsymbol{y}^*$ so the row player will remain to play the ϵ-Nash equilibrium (due to the stability property).

With the stability property, we prove that LRCA can drive the row player to play an ϵ-Nash equilibrium, where ϵ can be chosen arbitrarily small. However, in situations where the stability property does not hold (i.e., $\boldsymbol{A}\boldsymbol{y}^* \neq [v, v, \ldots, v]$ or the row player follows μ-regret algorithms with constant step size), we need different analyses for LRCA. In the following sections, we provide refined analyses for the LRCA algorithm with respect to specific algorithms followed by the row player.

3.2. Last Round Convergence under MWU/LMWU

One of the most well-studied no-regret algorithms in the game theory literature is the multiplicative weights update (MWU) method, which can be defined as follows:

Definition 6 (Freund and Schapire (1999)) *Let $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ be a sequence of mixed strategies played by the column player. The row player is said to follow the MWU algorithm if strategy $\boldsymbol{x}_{t+1}$ is updated as follows:*

$$\boldsymbol{x}_{t+1}(i) = \boldsymbol{x}_t(i)\frac{e^{-\mu_t \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_t}}{Z_t},\ i \in \{1, \ldots n\},$$
$$\text{where } \begin{cases} Z_t = \sum_{i=1}^n \boldsymbol{x}_t(i) e^{-\mu_t \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_t}, \mu_t \in [0, \infty) \text{ is a parameter,} \\ \boldsymbol{e}_i,\ i \in \{1, \ldots, n\}, \text{ is the unit-vector with } 1 \text{ at the } i\text{th component.} \end{cases}$$

Bailey and Piliouras (2018) proved that if both players follow MWU then in the case of interior minimax equilibrium, the strategies will move away from the equilibirium and towards the boundary (e.g., the blue trajectory in Figure 1).

A variant of MWU is the Linear Multiplicative Weight Update (LMWU), which is also a no-regret algorithm with suitable step size:

Definition 7 *The row player is said to play the LMWU if the row player updates the strategy as follows:*

$$\boldsymbol{x}_{t+1}(i) = \frac{\boldsymbol{x}_t(i)(1 - \mu_t \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_t(j)(1 - \mu_t \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_t)}\ \forall i \in \{1, \ldots n\}.$$

In this subsection, we prove that Algorithm 1 (LRCA) played by the column player will lead to last round convergence in the case of MWU/LMWU. The following lemma shows that the relative entropy between strategy of the row player and the minimax equilibrium is non-increasing.

Lemma 8 *Assume that the row player follows the MWU/LMWU algorithm with a non-increasing step size μ_t such that there exists $t' \in \mathbb{N}$ with $\mu_{t'} \leq \frac{1}{3}$. If the column player follows LRCA with $\beta \geq 2$ then*

$$RE\left(\boldsymbol{x}^* \| \boldsymbol{x}_{2k-1}\right) - RE\left(\boldsymbol{x}^* \| \boldsymbol{x}_{2k+1}\right) \geq \frac{1}{2}\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) \;\; \forall k \in \mathbb{N}: \;\; 2k \geq t',$$

where RE denotes the relative entropy, which is defined in Definition 2.

This Lemma can be used to prove the following result:

Theorem 9 *Let $\boldsymbol{A}$ be an $n \times m$ non-zero matrix with entries in $[0, 1]$. Assume that the row player follows the MWU/LMWU algorithm with a non-increasing step size μ_t such that $\lim_{T\to\infty} \sum_{t=1}^{\top} \mu_t = \infty$ and there exists $t' \in \mathbb{N}$ with $\mu_{t'} \leq \frac{1}{3}$. If the column player plays LRCA then there exists a minimax equilibrium $\bar{\boldsymbol{x}}^*$, such that $\lim_{t\to\infty} RE(\bar{\boldsymbol{x}}^* \| \boldsymbol{x}_t) = 0$ and thus $lim_{t\to\infty}\, \boldsymbol{x}_t = \bar{\boldsymbol{x}}^*$ almost everywhere and $lim_{t\to\infty}\, \boldsymbol{y}_t = \boldsymbol{y}^*$.*

Proof Let $\boldsymbol{x}^*$ be a minimax equilibrium strategy of the row player ($\boldsymbol{x}^*$ may not be unique). Since μ_t is a non-increasing step size, there exists t' such that $\mu_t \leq \frac{1}{3}$ for all $t \geq t'$. Following Lemma 8, for all $k \in \mathbb{N}$ such that $2k \geq t'$, we have

$$RE(\boldsymbol{x}^* \| \boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{2k-1}) \leq -\frac{1}{2}\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v). \tag{2}$$

Thus, the sequence of relative entropy $RE(\boldsymbol{x}^* \| \boldsymbol{x}_{2k-1})$ is non-increasing for all $k \geq \frac{t'}{2}$. As the sequence is bounded below by 0, it has a limit for any minimax equilibrium strategy $\boldsymbol{x}^*$. Since t' is a finite number and $\sum_{t=1}^{\infty} \mu_t = \infty$, we have $\sum_{t=t'}^{\infty} \mu_t = \infty$. Thus,

$$\lim_{T\to\infty} \sum_{k=\left\lceil \frac{t'}{2} \right\rceil}^{T} \mu_{2k} = \infty.$$

We will prove that $\forall \epsilon > 0, \;\; \exists h \in \mathbb{N}$ such that following LRCA for the column player and MWU/LMWU algorithm for the row player, the row player will play strategy $\boldsymbol{x}_h$ at round h and $f(\boldsymbol{x}_h) - v \leq \epsilon$. In particular, we prove this by contradiction. That is, suppose that $\exists \epsilon > 0$ such that $\forall h \in \mathbb{N}, \;\; f(\boldsymbol{x}_h) - v > \epsilon$. Then $\forall k \in \mathbb{N}$,

$$\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) = \frac{(f(\boldsymbol{x}_{2k-1}) - v)^2}{\beta} > \frac{e^2}{\beta}.$$

Let k vary from $\left\lceil \frac{t'}{2} \right\rceil$ to T in equation (2). By summing over k, we obtain:

$$\begin{aligned} RE(\boldsymbol{x}^* \| \boldsymbol{x}_{2T+1}) &\leq RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t'}) - \frac{1}{2} \sum_{k=\left\lceil \frac{t'}{2} \right\rceil}^{T} \mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) \\ &\leq RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t'}) - \frac{1}{2}\frac{e^2}{\beta} \sum_{k=\left\lceil \frac{t'}{2} \right\rceil}^{T} \mu_{2k}. \end{aligned}$$

Since $\lim_{T\to\infty}\sum_{k=\lceil\frac{t'}{2}\rceil}^{T}\mu_{2k}=\infty$ and $RE(\boldsymbol{x}^*\|\boldsymbol{x}_{T+1})\geq 0$, which contradicts our assumption about $\forall h\in\mathbb{N},\ f(\boldsymbol{x}_h)-v>\epsilon$.

Now, we take a sequence of $\epsilon_k>0$ such that $\lim_{k\to\infty}\epsilon_k=0$. Then for each k, there exists $\boldsymbol{x}_{t_k}\in\Delta_n$ such that $v\leq f(\boldsymbol{x}_{t_k})\leq v+\epsilon_k$. As Δ_n is a compact set and $\boldsymbol{x}_{t_k}$ is bounded then following the Bolzano-Weierstrass theorem, there is a convergence subsequence $\boldsymbol{x}_{\bar{t}_k}$. The limit of that sequence, $\bar{\boldsymbol{x}}^*$, is a minimax equilibrium strategy of the row player (since $f(\bar{\boldsymbol{x}}^*)=f(\lim_{k\to\infty}\boldsymbol{x}_{\bar{t}_k})=\lim_{k\to\infty}f(\boldsymbol{x}_{\bar{t}_k})=v$). Combining with the fact that $RE(\bar{\boldsymbol{x}}^*\|\boldsymbol{x}_{2k-1})$ is non-increasing for $k\geq\left\lceil\frac{t'}{2}\right\rceil$ and $RE(\bar{\boldsymbol{x}}^*\|\bar{\boldsymbol{x}}^*)=0$, we have $\lim_{k\to\infty}RE(\bar{\boldsymbol{x}}^*\|\boldsymbol{x}_{2k-1})=0$. We also note that

$$\begin{aligned}RE(\bar{\boldsymbol{x}}^*\|\boldsymbol{x}_{2k})-RE(\bar{\boldsymbol{x}}^*\|\boldsymbol{x}_{2k-1})&=\mu_{2k-1}\bar{\boldsymbol{x}}^{*\top}\boldsymbol{A}\boldsymbol{y}_{2k-1}+\log\left(\sum_{i=1}^{n}\boldsymbol{x}_{2k-1}(i)e^{-\mu_{2k-1}\boldsymbol{e}_i^\top\boldsymbol{A}\boldsymbol{y}^*}\right)\\&\leq\mu_{2k-1}v+\log\left(\sum_{i=1}^{n}\boldsymbol{x}_{2k-1}(i)e^{-\mu_{2k-1}v}\right)=0,\end{aligned}$$

following the fact that $\boldsymbol{x}^{*\top}\boldsymbol{A}\boldsymbol{y}\leq v$ for all $\boldsymbol{y}\in\Delta_m$ and $\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{y}^*\geq v$ for all $\boldsymbol{x}\in\Delta_n$. Thus, we have $\lim_{k\to\infty}RE(\bar{\boldsymbol{x}}^*\|\boldsymbol{x}_{2k})=0$ as well. Subsequently, $\lim_{t\to\infty}RE(\bar{\boldsymbol{x}}^*\|\boldsymbol{x}_t)=0$, which concludes the proof. ■

The optimal step size α_t in the case of MWU is $\alpha_t=\frac{f(\boldsymbol{x}_{t-1})-v}{\mu_t f(\boldsymbol{x}_{t-1})}$. However, in order to make LRCA robust against other algorithms of the row player, we choose the step size as shown in the algorithm. In LRCA 1, if we have $f(\boldsymbol{x}_t)-v\leq\epsilon$, then

$$min_{\boldsymbol{x}\in\Delta_n}\boldsymbol{x}^\top A\boldsymbol{y}_t\geq(1-\alpha_t)v\geq(1-\epsilon)v\implies v-min_{\boldsymbol{x}\in\Delta_n}\boldsymbol{x}^\top A\boldsymbol{y}_t\leq\epsilon.$$

It is easy to show that these inequalities imply $(\boldsymbol{x}_t,\boldsymbol{y}_t)$ is 2ϵ-nash equilibrium. Follow Lemma 8 in the case of constant learning rate $\mu_t=\mu$, we have the complexity of the algorithm in order to achieve ϵ-nash equilibrium is $\mathcal{O}(\frac{\log(n)/\mu}{\epsilon^2})$.

3.3. Last Round Convergence under FTRL

We now consider a more general form of no-regret algorithms, namely Follow the Regularized Leader (e.g., see Abernethy et al. (2008)).

Definition 10 *The row player is said to play the FTRL with σ-strongly convex regularizer: $F(\boldsymbol{x})$ if the row player updates the strategy as follows:*

$$\boldsymbol{x}_t=\operatorname*{argmin}_{\boldsymbol{x}\in\Delta_n}G_t(\boldsymbol{x})=\boldsymbol{x}^\top\left(\sum_{i=1}^{t-1}\boldsymbol{A}\boldsymbol{y}_i\right)+\frac{1}{\mu}F(\boldsymbol{x}).$$

FTRL covers a large set of well-known no-regret algorithms. For instance, if the negative entropy function is used as the regularizer, then FTRL results in a fixed step-size Multiplicative Weight Update. In the case of Euclidean regularizer, the FTRL becomes the famous Online Mirror Descent with lazy projection (e.g. see Shalev-Shwartz et al. (2012)). We now have an analysis about last round convergence when play against the general FTRL :

Theorem 11 *Assume that the row player follows the FTRL with σ-strongly convex regularizer: $F(\boldsymbol{x})$ with fixed step such that $\mu \leq 1$ and $\sigma \geq 1$. Then if the column player follows the Algorithm 1 (LRCA) with $\beta \geq n^2$, there will be last round convergence to the minimax equilibrium.*

Proof Let $\boldsymbol{x}^*$ be a minimax equilibrium of the row player. Denote $H_t(\boldsymbol{x}^*) = G_t(\boldsymbol{x}^*) - G_t(\boldsymbol{x}_t)$, following properties of strongly convex function we have:

$$H_t \geq \frac{\sigma}{2\mu}\|\boldsymbol{x}^* - \boldsymbol{x}_t\|^2.$$

Thus, if $H_t(\boldsymbol{x}^*)$ converges to 0 then we have $\boldsymbol{x}_t$ converges to $\boldsymbol{x}^*$. We will prove that

$$H_{t-1}(\boldsymbol{x}^*) - H_{t+1}(\boldsymbol{x}^*) \geq \frac{(f(\boldsymbol{x}_{t-1}) - v)^2}{2n^2} \quad \forall t = 2k.$$

From the definition of H_t we have:

$$\begin{aligned} H_{t-1}(\boldsymbol{x}^*) - H_{t+1}(\boldsymbol{x}^*) &= (G_{t+1}(\boldsymbol{x}_{t+1}) - G_{t-1}(\boldsymbol{x}_{t-1})) - (G_{t+1}(\boldsymbol{x}^*) - G_{t-1}(\boldsymbol{x}^*)) \\ &= (G_{t-1}(\boldsymbol{x}_{t+1}) - G_{t-1}(\boldsymbol{x}_{t-1}) + {\boldsymbol{x}_{t+1}}^\top \boldsymbol{A}(\boldsymbol{y}_{t-1} + \boldsymbol{y}_t)) - {\boldsymbol{x}^*}^\top \boldsymbol{A}(\boldsymbol{y}_{t-1} + \boldsymbol{y}_t) \\ &\geq \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 + {\boldsymbol{x}_{t+1}}^\top \boldsymbol{A}(\boldsymbol{y}_{t-1} + \boldsymbol{y}_t) - {\boldsymbol{x}^*}^\top \boldsymbol{A}(\boldsymbol{y}_{t-1} + \boldsymbol{y}_t), \end{aligned} \tag{3a}$$

where the last inequality derives from strongly convex property of G_{t-1}. We note that as $\boldsymbol{x}^* = \operatorname{argmin}_{\boldsymbol{x} \in \Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^*$, the following inequality holds

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^* \geq {\boldsymbol{x}^*}^\top \boldsymbol{A}\boldsymbol{y}^* = v \; \forall \boldsymbol{x} \in \Delta_n.$$

Plug it in the inequality (3a) and note that $\boldsymbol{y}_{t-1} = \boldsymbol{y}^*$ for an even t, then we have:

$$\begin{aligned} H_{t-1}(\boldsymbol{x}^*) - H_{t+1}(\boldsymbol{x}^*) &\geq \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 + (\boldsymbol{x}_{t+1} - \boldsymbol{x}^*)^\top \boldsymbol{A}\boldsymbol{y}_t \\ &= \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 + (\boldsymbol{x}_{t+1} - \boldsymbol{x}^*)^\top ((1-\alpha_t)\boldsymbol{y}^* + \alpha_t \boldsymbol{e}_t) && \text{(4a)} \\ &\geq \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 + \alpha_t (\boldsymbol{x}_{t+1} - \boldsymbol{x}^*)^\top \boldsymbol{A}\boldsymbol{e}_t && \text{(4b)} \\ &= \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 + \alpha_t (\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1})^\top \boldsymbol{A}\boldsymbol{e}_t + \alpha_t (\boldsymbol{x}_{t-1} - \boldsymbol{x}^*)^\top \boldsymbol{A}\boldsymbol{e}_t \\ &\geq \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 - \alpha_t \|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\| \; \|\boldsymbol{A}\boldsymbol{e}_t\|_* + \alpha_t (f(\boldsymbol{x}_{t-1}) - v). && \text{(4c)} \end{aligned}$$

Inequalities (4a,4b) come from the definition of $\boldsymbol{y}_t$. We have the inequalies (4c) as the result of,

$$a^\top b \leq \|a\|\|b\|_*,$$

where $\|\;\|_*$ denotes the dual norm. For vector a such that $\{a | 0 < a(i) \leq 1 \; \forall i \in [n]\}$ we have:

$$\|a\|_* \leq n.$$

Substitute this into the inequalities (4c) we have:

$$\begin{aligned} H_{t-1}(\boldsymbol{x}^*) - H_{t+1}(\boldsymbol{x}^*) &\geq \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 - n\alpha_t\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\| \ + \alpha_t(f(\boldsymbol{x}_{t-1}) - v) \\ &= \left(\sqrt{\frac{\sigma}{2\mu}}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\| - \frac{n\alpha_t}{2\sqrt{\frac{\sigma}{2\mu}}}\right)^2 + \alpha_t(f(\boldsymbol{x}_{t-1}) - v) - \frac{n^2\alpha_t^2\mu}{2\sigma} \\ &\geq \alpha_t(f(\boldsymbol{x}_{t-1}) - v) - \frac{n^2\alpha_t^2\mu}{2\sigma} \geq \alpha_t(f(\boldsymbol{x}_{t-1}) - v) - \frac{n^2\alpha_t^2}{2}. \end{aligned} \tag{5a}$$

Now, from LRCA 1 we have

$$\alpha_t = \frac{f(\boldsymbol{x}_{t-1} - v)}{n^2},$$

then inequality (5a) implies:

$$H_{t-1}(\boldsymbol{x}^*) - H_{t+1}(\boldsymbol{x}^*) \geq \frac{\alpha_t}{2}(f(\boldsymbol{x}_{t-1}) - v) = \frac{(f(\boldsymbol{x}_{t-1}) - v)^2}{2n^2} \geq 0 \ \forall t \text{ even}. \tag{6}$$

Follow the same argument in the proof of Theorem 9, we have the last round convergence result. ■

Note that we only need an upper bound for μ and a lower bound for σ in order to prove the Theorem 11. The FTRL with negative entropy regularizer becomes the MWU with constant step size μ. However, when μ varies in each update, then the two algorithms can be significantly different and thus the analysis in Theorem 9 is necessary. From the analysis of Theorem 11, the complexity of the algorithm in order to achieve ϵ-nash equilibrium is $\mathcal{O}(\frac{n^2}{\epsilon^2})$.

3.4. Convergence with Minimax Equilibrium Estimation

It is well-known that if both players follow a no-regret algorithm, then the average strategy will converge to a minimax equilibirium (Cesa-Bianchi and Lugosi, 2006). Bailey and Piliouras (2019) considered a more interesting setting where both players use a constant step size gradient algorithm(i.e., algorithms with a constant regret). They proved that in the case of 2×2 matrix $\boldsymbol{A}$, there will be average convergence to minimax equilibrium. Further, their experimental results suggest that the result hold true for a every size of matrix $\boldsymbol{A}$. In this section, we consider a symmetric game in which the row player follows the Multiplicative Weight Update Algorithm. Without the knowledge of the matrix $\boldsymbol{A}$, the column player first plays a no-regret algorithm and collect the historical average strategy: an estimation of $\boldsymbol{y}^*$. After having the estimation, the column player then follows the Algorithm 1. We prove that the strategies of the row and column player will converge to an arbitrarily small ball containing the minimax equilibrium.

Theorem 12 *Assume that the row player follows the MWU algorithm with a fixed step size $\mu > 0$. For any $\lambda > 0$, there exists $\epsilon > 0$ such that if the column player follows LRCA with the approximation of $\boldsymbol{y}^*$, v as $\bar{\boldsymbol{y}}$, $\bar{v}$ and $\min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\bar{\boldsymbol{y}} > v - \epsilon$ with $v + \epsilon > \bar{v} > v - \epsilon$, then there exist T and such that for every $t > T$, there is a minimax equilibrium $\boldsymbol{x}^*$ such that $RE(\boldsymbol{x}^*\|\boldsymbol{x}_t) < \lambda$.*

Now, we prove that the LRCA algorithm is a no-dynamic regret algorithm under mild conditions.

4. No-Dynamic Regret Algorithm

In this section, we first show that if the column player wants to achieve both the no-regret and stability properties, then the row player's strategy needs to converge to the minimax equilibrium. We then show that LRCA is a no-dynamic regret algorithm for the column player when the row player follows the aforementioned no-regret algorithms. In a general case, we suggest a method to combine our LRCA algorithm with another no-regret algorithm (such that Adahedge (De Rooij et al., 2014)) so that the new algorithm will still have no-regret property against random sequences of the row player while maintaining no-dynamic regret in the specific situation.

Lemma 13 *Suppose that the row player follows a common no-regret algorithm such as MWU, OMD, FTRL, LMWU or OMWU. Then, the column player cannot achieve last round convergence and the no-regret property if the row player's strategy does not converge to a minimax equilibrium of the game.*

Our algorithm LRCA satisfies the sufficient condition in Lemma 13. Next, we prove the no-dynamic regret property of the algorithm, clarifying the design of LRCA.

Theorem 14 *Assume that the row player follows the above-mentioned no-regret type algorithms: MWU/LMWU, FTRL. If there exists a fully mixed minimax strategy for the row player, then by following LRCA, the column player will achieve the no-dynamic regret property with the dynamic regret satisfying* $R_T \leq DR_T = \mathcal{O}\left(\sqrt{\log(n)}T^{3/4}\right)$. *Furthermore, in the case the row player uses a constant learning rate* μ, *we have* $DR_T = \mathcal{O}(\frac{n}{\sqrt{\mu}}T^{1/2})$.

In the case of row player uses constant learning rate, LRCA achieves the average dynamic regret of $\mathcal{O}(T^{-1/2})$, better than state of the art no-regret algorithms which obtain the same average but in the normal regret R_T.

In the general case where the column player does not know whether the row player use the following algorithm to achieve no-regret property in any situations while maintaining the no-dynamic regret property against no-regret algorithm of the row player: The idea is to put LCRA on top of another no-regret algorithm. When the regret of LCRA exceeds a certain threshold, we swap to the chosen algorithm. If the row player follows a no-regret algorithm then the LRCA regret will never exceed the threshold; thus we will have no-dynamic regret. By doing that, the column player sacrifices the optimal rate of no-regret in the worst case in order to achieve a much better no-dynamic regret in the case the row player follows a no-regret algorithm. The new Algorithm 2 will have the regret $R_T = O(\sqrt{\log(n)}T^{3/4})$ against random sequence strategies of the row player while maintain no-dynamic regret against the no-regret algorithm of the row player.

5. Discussions

The main focus of this paper is on the last convergence property in no-regret algorithm dynamics. By considering asymmetric goals of the players, we prove that there is a natural method to achieve last round convergence, which is proven not to hold in the old symmetric setting. This will strengthen the study in no-regret algorithms in the theoretical community and open to more interesting problems in which last round convergence can be achieved (e.g., see Dinh et al. (2020), Daskalakis and Panageas (2018)). With recent understanding about no-regret algorithms in online learning, more and more researchers try to move away from average convergence of strategies towards last round convergence

Algorithm 2: Combination of LRCA and Adahedge algorithm

Input: Current iteration t, past feedback $\boldsymbol{x}_{t-1}^{\top}\boldsymbol{A}$ of the row player, total regret up to time t: R_t
Output: Strategy $\boldsymbol{y}_t$ for the column player
if $R_t \leq \sqrt{\log(n)}t^{3/4}$ **then**
| Follow the Algorithm 1 (LRCA)
else
| Follow Adahedge algorithm De Rooij et al. (2014) onwards
end

of strategy. This will give us an idea on how the Nash equilibrium can arise naturally in a dynamic. Furthermore, the last round convergence will give the system a stable status so that a system designer can exploit this nice property.

References

Jacob Abernethy, Elad E Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory, COLT 2008*, pages 263–273, 2008.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

James Bailey and Georgios Piliouras. Fast and furious learning in zero-sum games: vanishing regret with non-vanishing step sizes. In *Advances in Neural Information Processing Systems*, pages 12977–12987, 2019.

James P Bailey and Georgios Piliouras. Multiplicative weights update in zero-sum games. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 321–338, 2018.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Constantinos Daskalakis and Ioannis Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. *arXiv preprint arXiv:1807.04252*, 2018.

Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. Near-optimal no-regret algorithms for zero-sum games. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 235–254. SIAM, 2011.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research*, 15(1):1281–1316, 2014.

Le Cong Dinh, Nick Bishop, and Long Tran-Thanh. Exploiting no-regret algorithms in system design. *arXiv preprint arXiv:2007.11172*, 2020.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Tianyi Lin, Zhengyuan Zhou, Panayotis Mertikopoulos, and Michael I Jordan. Finite-time last-iterate convergence for multi-agent learning in games. *arXiv preprint arXiv:2002.09806*, 2020.

Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *AISTATS*, pages 525–533, 2011.

Panayotis Mertikopoulos, Christos Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2703–2717. SIAM, 2018.

Dragoslav S Mitrinovic and Petar M Vasic. *Analytic inequalities*, volume 61. Springer, 1970.

J v Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Appendix A. Appendix A: proofs

Proof [Lemma 8]

First, we provide the proof in case of MWU. Following the Definition 2 of relative entropy we have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&= (RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k})) + (RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1})) \\
&= \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k+1}(i)}\right) - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k}(i)}\right)\right) + \\
&\left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k}(i)}\right) - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k-1}(i)}\right)\right) \\
&= \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}_{2k}(i)}{\boldsymbol{x}_{2k+1}(i)}\right)\right) + \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}_{2k-1}(i)}{\boldsymbol{x}_{2k}(i)}\right)\right).
\end{aligned}$$

Following the update rule of the multiplicative weights update algorithm in Definition 3.1 we have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&= \left(\mu_{2k}\boldsymbol{x}^{*\top}\boldsymbol{A}\boldsymbol{y}_{2k} + \log(Z_{2k})\right) + \left(\mu_{2k-1}\boldsymbol{x}^{*\top}\boldsymbol{A}\boldsymbol{y}_{2k-1} + \log(Z_{2k-1})\right) \\
&\leq \left(\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k}(i) e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right)\right) + (\mu_{2k-1}v + \log(Z_{2k-1})) \qquad (7a) \\
&= \left(\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k-1}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k-1}} e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right) - \log(Z_{2k-1})\right) \\
&+ (\mu_{2k-1}v + \log(Z_{2k-1})),
\end{aligned}$$

where Inequality (7a) is due to the fact that $\boldsymbol{x}^{*\top}\boldsymbol{A}\boldsymbol{y} \leq v \; \forall \boldsymbol{y} \in \Delta_m$. Thus,

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \left(\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k-1}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}^*} e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right)\right) + \mu_{2k-1}v \\
&\leq \left(\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k-1}v} e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right)\right) + \mu_{2k-1}v \qquad (8a) \\
&= \mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right),
\end{aligned}$$

where Inequality (8a) is the result of the inequality:

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^* \geq v \quad \forall \boldsymbol{x} \in \Delta_n.$$

Now, using the update rule of Algorithm 1 (LRCA)

$$\boldsymbol{y}_{2k} = (1 - \alpha_{2k})\boldsymbol{y}^* + \alpha_{2k}\boldsymbol{e}_{2k},$$

we then have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \mu_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right) \\
&= \mu_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}((1-\alpha_{2k})\boldsymbol{y}^* + \alpha_{2k}\boldsymbol{e}_{2k})}\right) \\
&\leq \mu_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k}((1-\alpha_{2k})v + \boldsymbol{e}_i^\top \boldsymbol{A}(\alpha_{2k}\boldsymbol{e}_{2k}))}\right). \qquad (9a)
\end{aligned}$$

The Inequality (9a) holds as:

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^* \geq v \quad \forall \boldsymbol{x} \in \Delta_n.$$

This leads to

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \mu_{2k}\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k}\alpha_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{e}_{2k}}\right) \\
&\leq \mu_{2k}\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i)(1 - (1 - e^{-\mu_{2k}\alpha_{2k}})\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{e}_{2k}\right) \qquad (10a) \\
&= \mu_{2k}\alpha_{2k} v + \log\left(1 - (1 - e^{-\mu_{2k}\alpha_{2k}})\boldsymbol{x}_{2k-1}{}^\top \boldsymbol{A}\boldsymbol{e}_{2k}\right) \\
&\leq \mu_{2k}\alpha_{2k} v - (1 - e^{-\mu_{2k}\alpha_{2k}})\boldsymbol{x}_{2k-1}{}^\top \boldsymbol{A}\boldsymbol{e}_{2k} \qquad (10b) \\
&= \mu_{2k}\alpha_{2k} v - (1 - e^{-\mu_{2k}\alpha_{2k}}) f(\boldsymbol{x}_{2k-1}),
\end{aligned}$$

where Inequalities (10a, 10b) are due to

$$\beta^x \leq 1 - (1-\beta)x \quad \forall \beta \geq 0\ \boldsymbol{x} \in [0,1] \text{ and } \log(1-x) \leq -x \ \ \forall x < 1.$$

We can develop Inequality (10b) further as

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \mu_{2k}\alpha_{2k} v - \left(1 - e^{-\mu_{2k}\alpha_{2k}}\right) f(\boldsymbol{x}_{2k-1}) \\
&\leq \mu_{2k}\alpha_{2k} v - \left(1 - \left(1 - \mu_{2k}\alpha_{2k} + \frac{1}{2}(\mu_{2k}\alpha_{2k})^2\right)\right) f(\boldsymbol{x}_{2k-1}) \qquad (11a) \\
&= -\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) + \frac{1}{2}(\mu_{2k}\alpha_{2k})^2 f(\boldsymbol{x}_{2k-1}) \\
&\leq -\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) + \frac{1}{2}\mu_{2k}\alpha_{2k}\mu_{2k}\frac{f(\boldsymbol{x}_{2k-1}) - v}{f(\boldsymbol{x}_{2k-1})} f(\boldsymbol{x}_{2k-1}) \qquad (11b) \\
&\leq -\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) + \frac{1}{2}\mu_{2k}\alpha_{2k}\ (f(\boldsymbol{x}_{2k-1}) - v) \qquad (11c) \\
&= -\frac{1}{2}\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) \leq 0.
\end{aligned}$$

Here, Inequality (11a) is due to $e^x \leq 1 + x + \frac{1}{2}x^2 \;\; \forall \boldsymbol{x} \in [-\infty, 0]$, Inequality (11b) comes from the definition of α_t:

$$\alpha_t = \frac{f(\boldsymbol{x}_{t-1}) - v}{\beta},\; \beta \geq 2,\; f(\boldsymbol{x}_{2k-1}) \leq 1$$

and finally Inequality (11c) comes from the choice of k at the beginning of the proof, i.e., $\mu_{2k} \leq 1$.

We now consider the LMWU case. From the step size assumption of LMWU algorithm, we have:

$$\exists t \in \mathbb{N} \text{ such that } \mu_t \leq \frac{1}{3} \text{ and } \sum_{i=t}^{\infty} \mu_i = \infty.$$

Using the update rule of LMWU in Definition 3.3 we obtain

$$\frac{\boldsymbol{x}_{m+1}(1)}{\boldsymbol{x}_m(1)} : \ldots : \frac{\boldsymbol{x}_{m+1}(n)}{\boldsymbol{x}_m(n)} = (1 - \mu_m \boldsymbol{e}_1{}^\top \boldsymbol{A}\boldsymbol{y}_m) : \ldots : (1 - \mu_m \boldsymbol{e}_n{}^\top \boldsymbol{A}\boldsymbol{y}_m)\; \forall m.$$

Take m equal t and $t-1$ and time the equations side by side we obtain

$$\begin{aligned}
&\frac{\boldsymbol{x}_{t+1}(1)}{\boldsymbol{x}_{t-1}(1)} : \frac{\boldsymbol{x}_{t+1}(2)}{\boldsymbol{x}_{t-1}(2)} : \ldots : \frac{\boldsymbol{x}_{t+1}(n)}{\boldsymbol{x}_{t-1}(n)} = (1 - \mu_t \boldsymbol{e}_1{}^\top \boldsymbol{A}\boldsymbol{y}_t)(1 - \mu_{t-1} \boldsymbol{e}_1{}^\top \boldsymbol{A}\boldsymbol{y}_{t-1}) : \\
&(1 - \mu_t \boldsymbol{e}_2{}^\top \boldsymbol{A}\boldsymbol{y}_t)(1 - \mu_{t-1} \boldsymbol{e}_2{}^\top \boldsymbol{A}\boldsymbol{y}_{t-1}) : \ldots : (1 - \mu_t \boldsymbol{e}_n{}^\top \boldsymbol{A}\boldsymbol{y}_t)(1 - \mu_{t-1} \boldsymbol{e}_n{}^\top \boldsymbol{A}\boldsymbol{y}_{t-1}) \\
&\implies \boldsymbol{x}_{t+1}(i) = \frac{\boldsymbol{x}_{t-1}(i)(1 - \mu_t \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t)(1 - \mu_{t-1} \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_{t-1})}{\sum_{j=1}^n \boldsymbol{x}_{t-1}(j)(1 - \mu_t \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_t)(1 - \mu_{t-1} \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_{t-1})} \quad \forall i \in 1, 2, \ldots n.
\end{aligned}$$

Note that for t is event, $\boldsymbol{y}_{t-1} = \boldsymbol{y}^*$ in LRCA-1 algorithm. For any i such that : $\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}^* = v$ we have:

$$\begin{aligned}
\frac{\boldsymbol{x}_{t+1}(i)}{\boldsymbol{x}_{t-1}(i)} &= \frac{(1 - \mu_{t-1} \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}^*)(1 - \mu_t \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_{t-1}(j)(1 - \mu_{t-1} \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}^*)(1 - \mu_t \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_t)} \\
&= \frac{(1 - \mu_{t-1} v)(1 - \mu_t \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_{t-1}(j)(1 - \mu_{t-1} \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}^*)(1 - \mu_t \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_t)} \\
&= \frac{(1 - \mu_t \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_t(j) \frac{1 - \mu_{t-1} \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}^*}{1 - \mu_{t-1} v}(1 - \mu_t \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_t)} \geq \frac{(1 - \mu_t \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_{t-1}(j)(1 - \mu_t \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_t)}.
\end{aligned}$$

The last inequality is due to $\boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}^* \geq v \;\; \forall j \in \{1, \ldots, n\}$.

We also have for any j such that : $\boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}^* > v$ then $\boldsymbol{x}^*(j) = 0$ for any minimax equilibrium strategy $\boldsymbol{x}^*$. Therefore, we have:

$$\begin{aligned}
RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t-1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t+1}) &= \sum_{i=1}^n \boldsymbol{x}^*(i) log\left(\frac{\boldsymbol{x}_{t+1}(i)}{\boldsymbol{x}_{t-1}(i)}\right) \\
&\geq \sum_{i=1}^n \boldsymbol{x}^*(i) \log\left(\frac{(1 - \mu_t \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_{t-1}(j)(1 - \mu_t \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_t)}\right) \\
&= \sum_{i=1}^n \boldsymbol{x}^*(i) \log\left(\frac{(1 - \mu_t \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t}\right).
\end{aligned}$$

Applying inequality $log(\boldsymbol{x}) \geq (x-1) - (x-1)^2 \; \forall x \geq 0.5$ to the above equation, we obtain

$$\begin{aligned}
&RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t-1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t+1}) \\
&\geq \sum_{i=1}^{n} \boldsymbol{x}^*(i) \left(\frac{(1 - \mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A} \boldsymbol{y}_t)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t} - 1 - \left(\frac{(1 - \mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A} \boldsymbol{y}_t)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t} - 1 \right)^2 \right) \\
&= \frac{\mu_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t - {\boldsymbol{x}^*}^\top \boldsymbol{A} \boldsymbol{y}_t)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t} - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \frac{\mu_t^2 (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t - {\boldsymbol{e}_i}^\top \boldsymbol{A} \boldsymbol{y}_t)^2}{(1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t)^2}.
\end{aligned}$$

Now, follow the Algorithm 1 (LRCA), we have: $\boldsymbol{y}_t = (1 - \alpha_t)\boldsymbol{y}^* + \alpha_t \boldsymbol{e}_t$. For j such that $\boldsymbol{e}_j^\top \boldsymbol{A} \boldsymbol{y}^* > v$, we have $\boldsymbol{x}^*(j) = 0$. We can simplify the above equation accordingly and use the Cauchy theorem to obtain

$$\begin{aligned}
&RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t-1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t+1}) \geq \\
&\frac{\mu_t (1 - \alpha_t)(\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}^* - v)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t} - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \frac{2\mu_t^2 (1 - \alpha_t)^2 (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}^* - v)^2}{(1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t)^2} \\
&+ \frac{\mu_t \alpha_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{e}_t - {\boldsymbol{x}^*}^\top \boldsymbol{A} \boldsymbol{e}_t)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t} - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \frac{2\mu_t^2 \alpha_t^2 (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{e}_t - \boldsymbol{e}_i^\top \boldsymbol{A} \boldsymbol{e}_t)^2}{(1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t)^2}.
\end{aligned} \tag{12}$$

For $\mu_t \leq \frac{1}{3}$ we have:

$$\frac{\mu_t (1 - \alpha_t)(\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}^* - v)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t} - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \frac{2\mu_t^2 (1 - \alpha_t)^2 (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}^* - v)^2}{(1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t)^2} \geq 0.$$

We also have:

$$\frac{(\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{e}_t - \boldsymbol{e}_i^\top \boldsymbol{A} \boldsymbol{e}_t)^2}{(1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t)^2} \leq \frac{1}{(1 - \mu_t)(1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t)}.$$

Follow the Inequality (12), we obtain

$$RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t-1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t+1}) \geq \frac{\mu_t \alpha_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{e}_t - {\boldsymbol{x}^*}^\top \boldsymbol{A} \boldsymbol{e}_t)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t} - \frac{2\mu_t^2 \alpha_t^2}{(1 - \mu_t)(1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t)}.$$

By definition of α_t in LRCA-1 algorithm

$$\alpha_t \leq \frac{\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{e}_t - v}{2} \leq \frac{\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{e}_t - {\boldsymbol{x}^*}^\top \boldsymbol{A} \boldsymbol{e}_t}{2},$$

along with $\mu_t \leq \frac{1}{3}$ we have:

$$\frac{1}{2} \frac{\mu_t \alpha_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{e}_t - {\boldsymbol{x}^*}^\top \boldsymbol{A} \boldsymbol{e}_t)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t} \geq \frac{2\mu_t^2 \alpha_t^2}{(1 - \mu_t)(1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t)}.$$

Thus, we have:

$$\begin{aligned}
RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t-1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t+1}) &\geq \frac{1}{2} \frac{\mu_t \alpha_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{e}_t - {\boldsymbol{x}^*}^\top \boldsymbol{A} \boldsymbol{e}_t)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t} \\
&\geq \frac{1}{2} \frac{\mu_t \alpha_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{e}_t - v)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{y}_t} \geq \frac{\mu_t \alpha_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A} \boldsymbol{e}_t - v)}{2} \geq 0 \; \forall t = 2k.
\end{aligned}$$

■

Proof [Theorem 5] We will prove the theorem by contradiction. Suppose there exists $\epsilon > 0$ such that:

$$f(\boldsymbol{x}_l) - v > \epsilon, \ \forall l \in \mathbb{N}.$$

Then, following the update rule of Algorithm 1 (LRCA) we have:

$$\boldsymbol{y}_{2k-1} = \boldsymbol{y}^* \ ; \ \alpha_{2k} = \frac{f(\boldsymbol{x}_{2k-1}) - v}{\beta} > \frac{\epsilon}{\beta}.$$

By the stability property, as $\boldsymbol{y}_{2k-1} = \boldsymbol{y}^*$, we then have: $\boldsymbol{x}_{2k-1} = \boldsymbol{x}_{2k}$. Following the update rule of Algorithm 1 (LRCA):

$$\begin{aligned}
\boldsymbol{x}_{2k}^T \boldsymbol{A} \boldsymbol{y}_{2k} &= \boldsymbol{x}_{2k-1}^T \boldsymbol{A} \left((1 - \alpha_{2k}) \boldsymbol{y}^* + \alpha_{2k} \mathbf{e}_{2k} \right) \\
&\geq (1 - \alpha_{2k}) v + \alpha_{2k} f(\boldsymbol{x}_{2k-1}) && \text{(13a)} \\
&> (1 - \alpha_{2k}) v + \alpha_{2k} (v + \epsilon) && \text{(13b)} \\
&\geq v + \frac{\epsilon^2}{\beta},
\end{aligned}$$

Where inequality (13a) is due to

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y}^* \geq v \ \forall \boldsymbol{x} \in \Delta_n,$$

and where inequality (13b) comes from the assumption that $f(\boldsymbol{x}_l) - v > \epsilon$. We then have:

$$\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}_t^T \boldsymbol{A} \boldsymbol{y}_t \geq \frac{v + \left(v + \frac{\epsilon^2}{\beta} \right)}{2} = v + \frac{\epsilon^2}{2\beta}.$$

We also note that, from the definition of the value of the game, we have:

$$\min_i \frac{1}{T} \sum_{t=1}^{T} \mathbf{e}_i^T \boldsymbol{A} \boldsymbol{y}_t = \min_i \mathbf{e}_i^T \boldsymbol{A} \frac{\sum_{t=1}^{T} \boldsymbol{y}_t}{T} \leq v.$$

Thus, we have:

$$\lim_{T \to \infty} \min_i \frac{1}{T} \sum_{t=1}^{T} \mathbf{e}_i^T \boldsymbol{A} \boldsymbol{y}_t - \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}_t^T \boldsymbol{A} \boldsymbol{y}_t \leq v - \left(v + \frac{\epsilon^2}{2\beta} \right) = -\frac{\epsilon^2}{2\beta},$$

contradicting to the definition of a no-regret algorithm:

$$\lim_{T \to \infty} \min_i \frac{1}{T} \sum_{t=1}^{T} \mathbf{e}_i^T \boldsymbol{A} \boldsymbol{y}_t - \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}_t^T \boldsymbol{A} \boldsymbol{y}_t = 0.$$

■

Proof [Theorem 12]

We first provide some lemma before proving the theorem.

Lemma 15 *Let $\boldsymbol{A}$ be a matrix a two-players zero-sum game with entries in [0,1]. For all $\lambda > 0$, there exists $\epsilon > 0$ such that:*

$$if \max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \leq v + \epsilon \implies \textit{There exists a minimax equilibrium } \boldsymbol{x}^* \textit{ such that } ||\boldsymbol{x} - \boldsymbol{x}^*|| < \lambda.$$

Proof Let denote $f(\boldsymbol{x}) = \max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}$. Consider a closed and bounded set:

$$S = \{\boldsymbol{x} \in \Delta_n |\ ||\boldsymbol{x} - \boldsymbol{x}^*|| \geq \lambda\ \forall \text{ equilibria points } \boldsymbol{x}^*\}.$$

$f(\boldsymbol{x})$ is a continuous function on the closed and bounded set S, therefore it achieves a minimum v' in S. Since the construction of S, we have $v' - v > 0$. Pick $0 < \epsilon < v' - v$, then

$$\forall \boldsymbol{x} \in \Delta_n\ max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \leq v + \epsilon \leq v' \implies x \notin S \implies \exists \boldsymbol{x}^* \text{such that} ||\boldsymbol{x} - \boldsymbol{x}^*|| < \lambda.$$

■

Lemma 16 *Let $\boldsymbol{x}$ be a point in the Δ_n. Then for every $\lambda > 0$, there exists $\epsilon > 0$ such that $\forall \boldsymbol{y} \in \Delta_n$*

$$||\boldsymbol{x} - \boldsymbol{y}|| < \epsilon \implies Re(\boldsymbol{x}||\boldsymbol{y}) \leq \lambda.$$

Proof For $\boldsymbol{x}_i = 0$, we have $\boldsymbol{x}_i \log(\frac{\boldsymbol{x}_i}{\boldsymbol{y}_i}) = 0$ so w.l.o.g, we assume $\boldsymbol{x}_i > 0\ \forall i \in [n]$. Let $\boldsymbol{x}_k = \min_{k \in [n]}$. Pick $0 < \epsilon < \boldsymbol{x}_k$ such that

$$\log(\frac{\boldsymbol{x}_k}{\boldsymbol{x}_k - \epsilon}) \leq \lambda.$$

With the assumption that $||\boldsymbol{x} - \boldsymbol{y}|| < \epsilon$, we have $y_i \geq \boldsymbol{x}_i - \epsilon$. Then,

$$RE(\boldsymbol{x}||\boldsymbol{y}) = \sum_{i=1}^{n} \boldsymbol{x}_i \log(\frac{\boldsymbol{x}_i}{\boldsymbol{y}_i}) \leq \sum_{i=1}^{n} \boldsymbol{x}_i \log(\frac{\boldsymbol{x}_i}{\boldsymbol{x}_i - \epsilon}) \leq \sum_{i=1}^{n} \boldsymbol{x}_i \log(\frac{\boldsymbol{x}_k}{\boldsymbol{x}_k - \epsilon}) = \log(\frac{\boldsymbol{x}_k}{\boldsymbol{x}_k - \epsilon}) \leq \lambda.$$

■

Following the Definition of relative entropy we have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&= (RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k})) + (RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k}) - RE(\boldsymbol{x}^*||2k-1)) \\
&= \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k+1}(i)}\right) - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k}(i)}\right)\right) + \\
&\quad \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k}(i)}\right) - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k-1}(i)}\right)\right) \\
&= \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}_{2k}(i)}{\boldsymbol{x}_{2k+1}(i)}\right)\right) + \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}_{2k-1}(i)}{\boldsymbol{x}_{2k}(i)}\right)\right).
\end{aligned}$$

Following the update rule of the multiplicative weights update algorithm we have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&= \left(\mu \boldsymbol{x}^{*\top} \boldsymbol{A}\boldsymbol{y}_{2k} + \log(Z_{2k})\right) + \left(\mu \boldsymbol{x}^{*\top} \boldsymbol{A}\boldsymbol{y}_{2k-1} + \log(Z_{2k-1})\right) \\
&\leq \left(\mu v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k}(i) e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right)\right) + (\mu v + \log(Z_{2k-1})) \qquad (14a) \\
&= \left(\mu v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k-1}} e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right) - \log(Z_{2k-1})\right) \\
&+ (\mu v + \log(Z_{2k-1}))\,,
\end{aligned}$$

where Inequality (14a) is due to the fact that $\boldsymbol{x}^{*\top} \boldsymbol{A}\boldsymbol{y} \leq v \; \forall \boldsymbol{y} \in \Delta_m$. Now, using the update rule of Algorithm (LRCA)

$$\boldsymbol{y}_{2k} = (1 - \alpha_{2k})\bar{\boldsymbol{y}} + \alpha_{2k}\boldsymbol{e}_{2k},$$

we then have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \left(\mu v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A}\bar{\boldsymbol{y}}} e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A}((1-\alpha_{2k})\bar{\boldsymbol{y}} + \alpha_{2k}\boldsymbol{e}_{2k})}\right)\right) + \mu v \\
&\leq \left(\mu v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu(v-\epsilon)} e^{-\mu(1-\alpha_{2k})(v-\epsilon) - \mu\alpha_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{e}_{2k}}\right)\right) + \mu v \qquad (15a) \\
&= \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu\alpha_{2k}\boldsymbol{e}_i^\top \boldsymbol{e}_{2k}}\right),
\end{aligned}$$

where Inequality (15a) is the result of the inequality:

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^* \geq v \quad \forall \boldsymbol{x} \in \Delta_n.$$

This leads to

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu\alpha_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{e}_{2k}}\right) \\
&\leq \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i)(1 - (1 - e^{-\mu\alpha_{2k}})\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{e}_{2k}\right) \qquad (16a) \\
&= \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v + \log\left(1 - (1 - e^{-\mu\alpha_{2k}})\boldsymbol{x}_{2k-1}{}^\top \boldsymbol{A}\boldsymbol{e}_{2k}\right) \\
&\leq \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v - (1 - e^{-\mu\alpha_{2k}})\boldsymbol{x}_{2k-1}{}^\top \boldsymbol{A}\boldsymbol{e}_{2k} \qquad (16b) \\
&= \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v - (1 - e^{-\mu\alpha_{2k}}) f(\boldsymbol{x}_{2k-1}),
\end{aligned}$$

where Inequalities (16a, 16b) are due to

$$\beta^x \leq 1 - (1 - \beta)x \quad \forall \beta \geq 0 \; \boldsymbol{x} \in [0, 1] \text{ and } \log(1 - x) \leq -x \;\; \forall x < 1.$$

We can develop Inequality (16b) further as

$$
\begin{aligned}
& RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
& \leq \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k}v - \left(1 - e^{-\mu\alpha_{2k}}\right) f(\boldsymbol{x}_{2k-1}) \\
& \leq \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k}v - \left(1 - \left(1 - \mu\alpha_{2k} + \frac{1}{2}(\mu\alpha_{2k})^2\right)\right) f(\boldsymbol{x}_{2k-1}) && \text{(17a)} \\
& = \epsilon(2\mu - \mu\alpha_{2k}) - \mu\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) + \frac{1}{2}(\mu\alpha_{2k})^2 f(\boldsymbol{x}_{2k-1}) \\
& \leq \epsilon(2\mu - \mu\alpha_{2k}) - \mu\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) + \frac{1}{2}\mu\alpha_{2k}\mu\frac{f(\boldsymbol{x}_{2k-1}) - \bar{v} + \epsilon}{f(\boldsymbol{x}_{2k-1})} f(\boldsymbol{x}_{2k-1}) && \text{(17b)} \\
& \leq \epsilon(2\mu - \mu\alpha_{2k}) - \frac{1}{2}\mu\alpha_{2k}(f_{2k-1} - v - 2\epsilon) && \text{(17c)} \\
& \leq \epsilon(2\mu - \mu\alpha_{2k}) - \frac{1}{2}\mu(f_{2k-1} - v)(f_{2k-1} - v - 2\epsilon).
\end{aligned}
$$

Here, Inequality (17a) is due to $e^x \leq 1 + x + \frac{1}{2}x^2 \;\; \forall x \in [-\infty, 0]$, Inequality (11b) comes from the definition of α_t:

$$\alpha_t = \frac{f(t-1) - \bar{v} + \epsilon}{f(t-1)},$$

and finally Inequality (11c) comes from the choice of k at the beginning of the proof, i.e., $\mu_{2k} \leq 1$. If

$$f_{2k-1} - v \geq 3\sqrt{\epsilon} \; and \; \epsilon \leq \frac{1}{4},$$

then we have

$$
\begin{aligned}
& RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
& \leq \epsilon(2\mu) - \frac{1}{2}3\sqrt{\epsilon}(2\sqrt{\epsilon}) \\
& \leq -\epsilon\mu < 0.
\end{aligned}
$$

Using lemma 15 and lemma 16, pick ϵ such that if $f_{2k-1} - v < 3\sqrt{\epsilon}$, then

$$RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) < \lambda_1 < \lambda - 3\epsilon\mu.$$

Since $RE()$ is non-negative, there exists k such that $f_{2k-1} - v < 3\sqrt{\epsilon}$. It leads to $RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) < \lambda - \epsilon\mu$. If $f_{2k+1} - v < 3\sqrt{\epsilon}$, then

$$RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) < \lambda_1 < \lambda - \epsilon\mu.$$

If $f_{2k+1} - v > 3\sqrt{\epsilon}$, then

$$RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) < RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) + 2\epsilon\mu < \lambda_1 + 2\epsilon\mu < \lambda - \epsilon\mu$$

$$RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+3}) < RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - \epsilon\mu < \lambda_1 + \epsilon\mu < \lambda - \epsilon\mu.$$

Following this process, we then have the K-L distance $RE(\boldsymbol{x}^*||\boldsymbol{x}_{2l-1}) < \lambda - \epsilon\mu < \lambda$ for all $l \geq k$.

For the even round, for all $l \geq k$ we have:

$$\begin{aligned} RE(\boldsymbol{x}^*||\boldsymbol{x}_{2l}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2l-1}) &= \mu {\boldsymbol{x}^*}^\top \boldsymbol{A}\boldsymbol{y}_{2l-1} + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2l-1}(i) e^{\mu {\boldsymbol{e}_i}^\top \boldsymbol{A}\bar{\boldsymbol{y}}}\right) \\ &\leq \mu v + log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2l-1}(i) e^{-\mu(v-\epsilon)}\right) \\ &= \mu\epsilon. \end{aligned}$$

This implies that

$$RE(\boldsymbol{x}^*||\boldsymbol{x}_{2l}) < RE(\boldsymbol{x}^*||\boldsymbol{x}_{2l-1}) + \mu\epsilon < (\lambda - \mu\epsilon) + \mu\epsilon = \lambda \; \forall \; l \geq k.$$

■

Proof [Lemma 13]

Suppose that the column player achieves both stability and no-regret property. The strategy of the column player will then converge, say to $\hat{\boldsymbol{y}}$. Following the property of common no-regret algorithms, the strategy of the row player will also converge to a single best response $\hat{\boldsymbol{x}}$ to $\hat{\boldsymbol{y}}$:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\Delta_n}{\operatorname{argmin}} \, \boldsymbol{x}^\top \boldsymbol{A}\hat{\boldsymbol{y}}.$$

Furthermore, since the strategy of the column player is no-regret, we must also have

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}\in\Delta_m}{\operatorname{argmax}} \, \hat{\boldsymbol{x}}^\top \boldsymbol{A}\boldsymbol{y}.$$

Therefore, by definition, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is a minimax equilibrium of the game. ■

Proof [Theorem 14]

We first prove the theorem in the case the row player follows the MWU/LMWU algorithm.

For the odd round $2k-1$, the dynamic regret of the column player at round $2k-1$ will satisfy

$$DR^{2k-1} = \max_{i\in 1,..m} \boldsymbol{x}_{2k-1}^\top \boldsymbol{A}\boldsymbol{e}_i - \boldsymbol{x}_{2k-1}^\top \boldsymbol{A}\boldsymbol{y}^* \leq f_{2k-1} - v.$$

For the even round $2k$, considering the existence of the fully mixed minimax equilibrium of the row player, we then have $\boldsymbol{A}\boldsymbol{y}^* = vI_1$ (I_1 is a vector of all 1 element) and thus $\boldsymbol{x}_{2k} = \boldsymbol{x}_{2k-1}$. Therefore $DR^{2k} \leq f_{2k-1} - v$.

Combining the case of odd and even round, we derive

$$DR_T \leq 2\sum_{k=1}^{T/2}(f_{2k-1} - v).$$

Now, following Lemma 8 in the case $n \geq 8$, we have

$$\begin{aligned} &\frac{1}{2}\mu_{2k}\frac{(f(\boldsymbol{x}_{2k-1}) - v)^2}{2} \leq RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) \\ \implies &\sum_{k=1}^{T/2} \mu_{2k}(f(\boldsymbol{x}_{2k-1}) - v)^2 \leq 4RE(\boldsymbol{x}^*||\boldsymbol{x}_1) \leq 4\log(n). \end{aligned}$$

Using the Cauchy–Schwarz inequality, we can then derive that

$$\sum_{k=1}^{T/2}(f(\boldsymbol{x}_{2k-1})-v) \leq 2\sqrt{\log(n)}\sqrt{\sum_{k=1}^{T/2}\frac{1}{\mu_{2k}}} \implies DR_T \leq 4\sqrt{\log(n)}\sqrt{\sum_{k=1}^{T/2}\frac{1}{\mu_{2k}}}.$$

If the row player follows the constant step size μ, then we have

$$DR_T \leq \frac{2\sqrt{2}\sqrt{\log(n)}}{\sqrt{\mu}}T^{1/2} = \mathcal{O}(\frac{n}{\sqrt{\mu}}T^{1/2}).$$

If the row player follows a decreasing step size $\mu_k = \sqrt{8\log(n)/k}$ (Cesa-Bianchi and Lugosi (2006)) to make the algorithm no-regret, then we have

$$DR_T \leq \log(n)^{1/4}T^{3/4} = \mathcal{O}(\sqrt{\log(n)}T^{3/4}).$$

Indeed, for any sequence of step size μ_k such that $\sum_{k=1}^{T/2}\frac{1}{\mu_{2k}} \leq \mathcal{O}(T^{3/2})$, the theorem holds.

We continue the proof in the case FTRL. W.l.o.g, assume that $\max_{\boldsymbol{x}\in\Delta_n} F(\boldsymbol{x}) = 1$. Following the proof of Theorem 11 we have

$$\sum_{k=1}^{T/2}(f(\boldsymbol{x}_{2k-1})-v)^2 \leq \frac{2n^2}{\mu}$$
$$\implies \sum_{k=1}^{T/2}(f(\boldsymbol{x}_{2k-1})-v) \leq \frac{1}{\sqrt{\mu}}T^{1/2}n.$$

Using the same argument as the case of MWU, we then have:

$$DR_T \leq \frac{2}{\sqrt{\mu}}T^{1/2}n = \mathcal{O}(\frac{n}{\sqrt{\mu}}T^{1/2}).$$

■