Adaptive Multiagent Reinforcement Learning with Non-positive Regret

Duong D. Nguyen[1(✉)], Langford B. White[1], and Hung X. Nguyen[2]

[1] School of Electrical and Electronic Engineering, The University of Adelaide,
Adelaide, SA 5005, Australia
{duong.nguyen,lang.white}@adelaide.edu.au

[2] Teletraffic Research Centre, The University of Adelaide,
Adelaide, SA 5005, Australia
hung.nguyen@adelaide.edu.au

Abstract. We propose a novel adaptive reinforcement learning (RL) procedure for multi-agent non-cooperative repeated games. Most existing regret-based algorithms only use positive regrets in updating their learning rules. In this paper, we adopt both positive and negative regrets in reinforcement learning to improve its convergence behaviour. We prove theoretically that the empirical distribution of the joint play converges to the set of correlated equilibrium. Simulation results demonstrate that our proposed procedure outperforms the standard regret-based RL approach and a well-known state-of-the-art RL scheme in the literature in terms of both computational requirements and system fairness. Further experiments demonstrate that the performance of our solution is robust to variations in the total number of agents in the system; and that it can achieve markedly better fairness performance when compared to other relevant methods, especially in a large-scale multiagent system.

Keywords: Multiagent systems · Reinforcement Learning · Game theory · Correlated equilibrium · No regret

1 Introduction

Reinforcement learning (RL) is a popular adaptive procedure used in distributed system and has been widely studied in artificial intelligence (AI) research areas (for a survey on recent developed RL algorithms refer to [1]). A RL procedure [2–6] does not require the agents to know anything about the entire environment, except their local information. Each agent learns about the environment by observing its own payoffs. Overtime, using only this information, it can rationally choose the best course of actions to maximise its objective utility (payoff). Under mild conditions of finite payoffs and of stationary environment, an RL procedure is guaranteed to converge to a set of stable equilibria.

Despite this very attractive property, RL procedure applying in multiagent settings suffers from two well-known problems of slow convergence and of convergence to sub-optimal equilibrium points, especially in a distributed system with

B.H. Kang and Q. Bai (Eds.): AI 2016, LNAI 9992, pp. 29–41, 2016.
DOI: 10.1007/978-3-319-50127-7_3

a very large number of agents [2]. Another challenge of RL-based algorithms is the inefficient of exploration. Since agents running RL procedure do not have a global knowledge of the whole system, they often require a high exploration times in order to converge to a stable equilibrium. In many application, these behaviours can result in undesirable outcomes [4,7].

This paper develops a new RL procedure that follows the regret-based principles [3,8] to overcome the disadvantage of slow speed and inefficient convergence of standard RL solutions. The notion of regret has been explored both in game theory and computer science [3,8–10]. Regret measures reflect how much worse in payoffs that an agent would experience if choosing other options instead of its current selection. In our problem formulation, we consider a multiagent non-cooperative repeated game with restricted information for the agents. Each agent only observe its own payoffs and know neither its payoff function nor the information on the other agents in the game. The goal of every agent is to guarantee no-regret in the long-term (average) payoffs.

Unlike most the existing regret-based algorithms that use only positive parts of regret measures to update the play probability and completely ignore negative regrets, we propose to use both positive and negative regrets to accelerate the convergence of the RL procedure. Our new approach is motivated by the observation that incorporation of negative regrets can help the agent to "explore" the environment more extensively as positive regrets decrease than the standard RL algorithm. The fact is that considering negative regrets can help agents make more "good" decisions by reducing unnecessary explorations on the actions that result in poor performances. Thus, more effective exploration has crucial impact on the convergence speed as well as the performance of the learning outcome.

However, since there is a negative impact on average performance by including more actions with negative regrets, our approach weights the impact of negative regrets on the probability distribution of actions in a manner that ensures (i) that actions with large (magnitude) negative regrets contribute less to the probability of choosing those actions than those with small (magnitude) negative regrets and (ii) that the contribution of negative regrets decreases to zero over time.

The main contribution of this paper are as follows:

1. *A Novel Adaptive Multiagent Reinforcement Learning Procedure:* We propose a novel fully distributed RL procedure that uses both positive and negative regret measures to improve convergence speed and fairness of the well-know regret-based RL procedure. We show that our solution is suitable for large-scale distributed multiagent systems.
2. *Our proof methodology:* We prove the convergence of our proposed procedure using differential inclusion (DI) technique. DI is a powerful theoretical framework that derived from the expected motion of a stochastic process. This paper demonstrates that the use of DI technique is particularly suitable to study the convergence behaviours of the regret based schemes and adaptive procedures in game theory, and provide a much more concise and extensible proof as compared to the classical approaches.

2 Background

This section reviews the background and notation used in this paper.

2.1 Game Model

We consider a game with A players denoted by the set $\{1, \ldots, A\}$ for some (finite) integer $A \geq 2$. Each player a has its set of actions (moves) $\mathcal{S}_a = \{1, \ldots, m\}$, where m is the number of action of player a. The set of all possible moves is the Cartesian product $\mathcal{S} = \Pi_{a=1}^{A} \mathcal{S}_a$. We view the game from the point of view of player one. Let $\mathcal{I} = S_1$ denote the set moves of player one and $\mathcal{L} = \mathcal{S} \setminus \mathcal{S}_1$ the set of moves of all other players. Denote by X, the set of all probability mass functions (pmf) on $\mathcal{I}$ and Y the set of pmf on $\mathcal{L}$. Let Z denote the set of pmf on $\mathcal{S}$, then $X \times Y$ is a subset of $\mathcal{Z}$ comprised of all pmf of the form $z = (x, y)$ where $x \in X$ and $y \in Y$, i.e. all pmf where the probability of the action of player one and the actions of all other players taken together, are statistically independent.

Let $U : \mathcal{S} \to \mathbb{R}$ denote the payoff achieved by player one when the overall action taken by all players is $s \in \mathcal{S}$. We represent a strategy in the form $s = (i, \ell)$ where i is the action of player one and ℓ is the action of all other players. We will consider the general formulation of game where users apply mixed strategies over the possible selection set $\mathcal{S}$. Under randomised actions with overall probability (pmf) $z \in Z$, the payoff obtained by player one is defined by extending the domain of definition of U to Z according to

$$U(z) = \sum\nolimits_{k \in \mathcal{S}} z(k)\, U(k). \tag{1}$$

Notice that U is a linear function. The multiagent game model then can be denoted by $\mathcal{G} = (\mathcal{A}, (\mathcal{S}_a)_{a \in \mathcal{A}}, (U_a)_{a \in \mathcal{A}})$.

2.2 Equilibrium States

In this paper, we are interested in a popular notion of rationality that generalises the Nash equilibrium called correlated equilibrium. It is an optimality concept introduced by Aumann [11]. It models possible correlation or co-ordination between players compared to the usual strategic equilibrium of Nash, where all players act independently. Correlated equilibrium is relevant to the probabilistic game, namely where strategies are determined probabilistically. Denote by ψ, a probability distribution defined in $\mathcal{S}$, the ψ is said to be a correlated equilibrium for the game $\mathcal{G}$ if for every player $a \in \mathcal{A}$, and for every pair of action $j, k \in \mathcal{S}^a$, it holds that

$$\sum\nolimits_{s \in \mathcal{S} : i = j} \psi(s)(U(k, \ell) - U(s)) \leq 0. \tag{2}$$

A correlated equilibrium results if each player does not benefit from choosing any other action, provided that all other players do likewise. When each player chooses their action independently of the other players, a correlated equilibrium is also a Nash equilibrium. We denote the set of correlated equilibria by CE.

2.3 Regret-Based Reinforcement Learning

A fully distributed procedure that can be used to reach the CE solution is the regret-based RL procedure [3]. The key idea of this method is to adjust the player's play probability proportional to the "regrets" for not having played other actions. Specifically, for any two actions $j \neq k \in \mathcal{I}$ at any time n, the regret of player one for not playing k is

$$[B_n]_{j,k} = \frac{1}{n} \sum_{t \leq n: i_t = j} U(k, \ell_t) - \frac{1}{n} \sum_{t \leq n: i_t = j} U(j, \ell_t). \tag{3}$$

This is the change in time average payoff that player one would have achieved if it substituted a given action j each time it was played in the past, with another action k. Since player one only knows his set of actions and his own payoffs, he cannot compute the first term. Thus, the regret in (3) needs to be replaced by an estimate that can be computed on the basic of the available information, as

$$[B_n]_{j,k} = \frac{1}{n} \sum_{t \leq n: i_t = k} \frac{p_t(j)}{p_t(k)} U(s_t) - \frac{1}{n} \sum_{t \leq n: i_t = j} U(s_t),$$

where, p_t denotes the play probabilities at time t, i.e., $p_t(k)$ is the probability of choosing k at time t and $U(s_t) = U(i_t, \ell_t)$ denotes the payoff at time t.

If $i_n = j$ is the action chosen by player one at time n, then the probability distribution that he chooses an action at time $n+1$ is defined recursively as [3]

$$p_{n+1}(k) = \begin{cases} \left(1 - \dfrac{\delta}{n^\gamma}\right) \min\left\{ \dfrac{[B_n]^+_{j,k}}{\mu}, \dfrac{1}{m} \right\} + \dfrac{\delta}{n^\gamma} \dfrac{1}{m}, & k \neq j, \\ 1 - \sum_{k' \neq j} p_{n+1}(k'), & k = j, \end{cases} \tag{4}$$

with the initial play probabilities at $t = 1$ uniformly distributed over the set of possible actions; $\mu > 2mG$ is a constant, m is the cardinality of the set $\mathcal{I}$ and G is an upper bound on $|U(s)|$ for all $s \in \mathcal{S}$; $0 < \delta < 1$ and $0 < \gamma < 1/4$. We use the notation $[B_n]^+_{j,k} := \max([B_n]_{j,k}, 0)$. By using $[B_n]^+_{j,k}$ in (4), the RL algorithm in [8] completely ignores negative regrets $[B_n]_{j,k} < 0$.

It is proven in [3] that if all players chooses their actions according to (4), the empirical distribution of all strategies played until time n, which is given by

$$z_n(s) = \frac{1}{n} \sum\nolimits_{t=1}^{n} \mathbb{1}_{\{s_t = s\}},$$

converges almost surely as $t \to \infty$ to the CE set of the game $\mathcal{G}$. Note that this does not imply convergence to a specific point on CE set, but that the solution approaches the CE set.

The main drawback of this standard regret-based reinforcement learning procedure is that although guaranteeing convergence to the set of CE, it often requires long convergence time and sometime converges to an undesirable equilibrium (i.e. poor fairness). These issues motivate the reinforcement learning with non-positive regret in the next section.

3 Algorithm

In this section, we describe our proposed multiagent reinforcement procedure.

3.1 Reinforcement Learning with Non-positive Regret

The RL procedure in Sect. 2.3 does not use any negative regrets in determining the probability of plays. However, as discussed in Sect. 1, negative regrets contain information that could improve the performance of the learning procedure. We propose to complement the regret-based RL in [3] by taking into account additional negative regrets in updating the learning rule. To determine the probability distribution of its action at the next stage $n+1$, agent uses both its positive and negative parts of the time average regrets as follow

$$p_{n+1}(k) = \begin{cases} \delta_n \dfrac{1}{m}, & \text{if } k \neq j \text{ and } [B_n]_{j,k} = 0 \\ (1-\delta_n)\ \dfrac{[B_n]^+_{j,k}}{\sum_k [B_n]^+_{j,k}} + \delta_n \dfrac{1}{m}, & \text{if } k \neq j \text{ and } [B_n]_{j,k} > 0 \\ (1-\delta_n)\ \dfrac{1}{n^\alpha} \dfrac{\left([B_n]^-_{j,k}\right)^{-1}}{\sum_k \left([B_n]^-_{j,k}\right)^{-1}} + \delta_n \dfrac{1}{m}, & \text{if } k \neq j \text{ and } [B_n]_{j,k} < 0 \\ 1 - \sum_{k' \neq j}\ p_{n+1}(k'), & \text{if } k = j \end{cases} \tag{5}$$

where $\delta_n = \delta/n^\gamma$ for $0 < \delta \ll 1$ and $0 < \gamma < 1/2$; and $0 < \alpha \leq 1$. We use the notation $[B_n]^-_{j,k} := \min([B_n]_{j,k}, 0)$.

Our main insight here is that the negative regrets should be included in the update procedure to ensure that when n is small the algorithm keep exploring different solutions, including the solution that yields negative regret, to speed up the convergence. However, as the algorithm progresses, the negative regrets reduce to zero and the positive regrets become the dominant factors in determining the playing probabilities. We prove that our new RL algorithm converges almost surely to the CE set and show in simulations that this learning strategy provides very fast convergence toward equilibrium states.

3.2 Discussion

We discuss in detail here the major differences between our solution and the standard regret-based RL approach [3]. The main novelty in our approach is in the formula to update the play probability.

(a) Firstly, we do not use a constant proportional factor μ as in (4), but normalise the vector of regret to get a probability vector. The reason for doing this is to avoid being dependent on the appropriate choice of some arbitrarily large enough parameter μ. As discussed in [3], a higher value of μ results in a smaller probability of switching and thus leads to a slower speed of convergence.

(b) Secondly, in our solution, not only positive regrets but also negative values are contributing to the update procedure of the player. In particular, the play probability is proportional to the positive regret and is proportional to the inverse of the negative regret. This choice of play probability allows the action that yields larger positive regret to get a higher probability to be selected in the next state, while the action that yields larger negative regrets to receive a lower probability to be used in the future.

(c) Thirdly, in the standard approach, it is difficult to determine an appropriate $0 < \delta < 1$ in (4). A large δ will lead the convergence to a large distance from the CE set hence lead to lower total utility. However, small δ means to discourage the exploration processes, and agents tend to perform the same action and thus will cause slow convergence. In our proposed approach, the choice of δ is much simpler: we only need to set $0 < \delta \ll 1$. A much smaller value of δ not only improves the convergence rate but also reduces the instability properties caused by inaccurate estimates of regrets in the standard RL solution. The key point here is that δ can be taken smaller to still obtain a similar amount of "exploration" due to the inclusion of the negative regret terms.

(d) Lastly, the negative regrets vanish in the play probability as the time step goes to infinity due to the inclusion of $1/n^{\alpha}$ in the play probability for negative regrets in (5). This means that the agent no longer considers the selection that yields negative regret after sufficiently exploring all the potential options. Using negative regrets after the exploration phase would reduce the achievable payoffs.

3.3 Convergence Analysis

Theorem 1. *If an agent (i.e. player one) uses the proposed procedure, its time average conditional regret is guaranteed to approach the set of non-positive regrets in the payoff space almost surely, provided that other agents do likewise.*

We now provide a brief overview of the proof. We use the differential inclusion (DI) framework in [12] to prove our Theorem. DI is a generalisation of ordinary differential equation that is particularly suitable to study the asymptotic trajectory of the iterative process in game-theoretic learning, especially when the information available to a player is "restricted". Standard approach in game theory such as Blackwell's approachability theorem used in [3,8], however, cannot be trivially extended to prove the convergence of the proposed algorithm and will require a significant number of additional steps to handle the modifications of the play probabilities p_n. The use of DI technique yields a considerably simpler and shorter proof as compared to the classical approach in [3].

Proof. Let $C : Z \to \mathbb{R}^{m \times m}$ be defined by

$$[C(z)]_{j,k} = \sum\nolimits_{\ell \in \mathcal{L}} z(j,\ell) \left(U(k,\ell) - U(j,\ell) \right),$$

which is the expected regret for player one when substituting action k for action j under the joint distribution z of actions. Suppose we consider player one playing some action i with probability one, then

$$\begin{aligned}[C(z^i)]_{j,k} &= \sum\nolimits_{\ell \in \mathcal{L}} \mathbb{1}_{\{i=j\}}\ y_\ell \left(U(k,\ell) - U(j,\ell)\right) \\ &= \mathbb{1}_{\{i=j\}} \left(U(k,y) - U(j,y)\right).\end{aligned}$$

Since player one cannot compute the first term as it only has access to the payoffs corresponding to actions it actually took, following [3], define an estimate of this term by

$$\tilde{U}(k,y)\ \mathbb{1}_{\{i=j\}} = \frac{p(j)}{p(k)}\ U(k,y)\ \mathbb{1}_{\{i=k\}}.$$

which is computed from the regrets associated with the alternative action k weighted proportional to the relative probabilities of player one choosing action j versus k when those actions were actually taken. The associated pseudo regret matrix at stage n is now

$$\tilde{C}_n(j,k) = \frac{p_n(j)}{p_n(k)}\ U(k,y_n)\ \mathbb{1}_{\{i_n=k\}} - U(j,y_n)\ \mathbb{1}_{\{i_n=j\}}.$$

Thus, we have

$$\begin{aligned}\mathbf{E}\left\{\tilde{C}_n(j,k)|h_{n-1}\right\} &= p_n(k)\frac{p_n(j)}{p_n(k)}U(k,y_n) - p_n(j)\ U(j,y_n) \\ &= p_n(j)\left(U(k,y_n) - U(j,y_n)\right) \\ &= \mathbf{E}\left\{C_n(j,k)|h_{n-1}\right\},\end{aligned}$$

where h_{n-1} is the action history of the game until stage $n-1$.

It can be seen that $C_n(j,k)$ and $\tilde{C}_n(j,k)$ are each bounded by $2mG/\delta_n$. The limit sets of the pair processes C_n and $\tilde{C}_n$ also coincide since they both have the same conditional expected values (see [3] for more details and discussions). Then Theorem 7.3 of [12] can be applied and thus the two processes exhibit the same asymptotic behaviour.

The average regret at stage n is thus a matrix B_n defined by

$$B_n(j,k) = \frac{1}{n}\sum_{t=1}^{n}\left[\frac{p_t(j)}{p_t(k)}U(k,y_t)\ \mathbb{1}_{\{i_t=k\}} - U(j,y_t)\ \mathbb{1}_{\{i_t=j\}}\right].$$

Hence, the discrete dynamics

$$\bar{B}_{n+1} - \bar{B}_n = \frac{1}{n+1}\ \left(B_{n+1} - \bar{B}_n\right)$$

is a discrete stochastic approximation of the DI

$$\dot{\mathbf{w}} \in N(\mathbf{w}) - \mathbf{w}\ \ (\text{with } w = B_n). \qquad (6)$$

Now for $j \neq k$, define the matrix sequence

$$[M_n]_{j,k} = \begin{cases} 0, & \text{if } [B_n]_{j,k} = 0 \\ \dfrac{[B_n]^+_{j,k}}{\sum_k [B_n]^+_{j,k}}, & \text{if } [B_n]_{j,k} > 0 \\ \dfrac{1}{n^\alpha} \dfrac{\left([B_n]^-_{j,k}\right)^{-1}}{\sum_k \left([B_n]^-_{j,k}\right)^{-1}}, & \text{if } [B_n]_{j,k} < 0 \end{cases} \tag{7}$$

We set $[M_n]_{j,j} = 1 - \sum_{k \neq j} [M_n]_{j,k}$, which takes value in $[0,1]$ by virtue of (7). Thus M_n is a transition probability matrix on $\mathcal{S}$. So there is a probability vector μ_n such that $M_n^T \mu_n = \mu_n$.

The "non-positive regret set" $D^1 \subset \mathbb{R}^{m \times m}$ for player one is defined by

$$D^1 = \left\{ g \in \mathbb{C}^{m \times m} : g(j,k) \leq 0, \forall (j,k) \right\}.$$

Evidently, D^1 is a closed, convex subspace of $\mathbb{R}^{m \times m}$. Define the Lyapunov function $P(w) = \frac{1}{2}\|w\|^2$, with $\nabla P(w) = w$. Then P satisfies the following properties and thus is a potential function for D^1:

- P is continuously differentiable;
- $P(w) = 0 \Leftrightarrow w \in D^1$;
- $\langle \nabla P(w), w \rangle > 0$ for all $w \notin D^1$.

Let $\varphi : \mathbb{R}^{m \times m} \to 2^X$ given by

$$\varphi(w) = \begin{cases} (1 - \delta_n)\ \mu(w) + \dfrac{\delta_n}{m}, & w \notin D^1 \\ X, & w \in D^1 \end{cases} \tag{8}$$

where $\mu(w)$ denotes a probability vector computed from the matrix $w = B_n$ according to the process above. Define a correspondence N on $\mathbb{R}^{m \times m} \setminus D^1$ by

$$N(w) = C(\varphi(w) \times Y)$$

so that φ is N-adapted, which means $N(w)$ contains all resulting average regrets.

According to Lyapunov theory, to prove the approachability of w to D^1, we need then to show that for any $w \in \mathbb{R}^{m \times m} \setminus D^1$ and some positive constant β,

$$\frac{d}{dt} P(w) = \langle \nabla P(w), \dot{w} \rangle \in \langle \nabla P(w), N(w) - w \rangle \leq -\beta P(w),$$

meaning that we need the following result

$$\langle \nabla P(w), \theta - w \rangle \leq -\beta P(w)$$

for all $\theta \in N(w)$ and some constant $\beta > 0$ (see [12] for details).

Suppose $w \notin D^1$, let $\theta = \mathbf{E}\left\{\tilde{C}(\varphi(w), y)|h_{n-1}\right\}$, with $y \in Y$, which means

$$[\theta]_{j,k} = \varphi_j(w)\ \left(U(k,y) - U(j,y)\right).$$

Then consider

$$\begin{aligned}
\langle \nabla P(w), \theta \rangle =& \sum\nolimits_{j,k}^{m} \nabla P_{jk}(w)\ \varphi_j(w)\ \left(U(k,y) - U(j,y)\right) \\
=& (1-\delta_n) \sum\nolimits_{j,k} \nabla P_{jk}(w)\ \mu_j(w)\ \left(U(k,y) - U(j,y)\right) \\
&+ \frac{\delta_n}{m} \sum\nolimits_{j,k} \nabla P_{jk}(w)\ \left(U(k,y) - U(j,y)\right) \\
=& (1-\delta_n) \sum\nolimits_{j} U(j,y) \left(\sum\nolimits_{k} \mu_k(w)\ \nabla P_{kj}(w) - \mu_j(w) \sum\nolimits_{k} \nabla P_{jk}(w)\right) \\
&+ \frac{\delta_n}{m} \sum\nolimits_{j,k} \nabla P_{jk}(w)\ \left(U(k,y) - U(j,y)\right). \qquad (9)
\end{aligned}$$

In the second line we substituted for $\varphi_j(w)$ from (8), and in the last line we collected together all terms containing $U(j,y)$.

Let $\mu_j(w)$ be such an invariant measure. Suppose that for every $j = 1, \ldots, m$, it holds that

$$\mu_j(w)\ \sum\nolimits_{k} \nabla P_{jk}(w) = \sum\nolimits_{k} \mu_k(w)\ \nabla P_{kj}(w),$$

then the first term in the sum in (9) is equal to zero. Therefore, noting that the payoff function $|U(.)|$ is bounded by G, we obtain

$$\begin{aligned}
\langle \nabla P(w), \theta \rangle &= \frac{\delta_n}{m} \sum\nolimits_{j,k} \nabla P_{jk}(w)\ \left(U(k,y) - U(j,y)\right) \\
&\leq ||\nabla P(w)||\ \frac{2G\delta_n}{m}. \qquad (10)
\end{aligned}$$

Next, using $P(w) = \|w\|^2/2$ and $\nabla P(w) = w$, it can be show that

$$\langle \nabla P(w), w \rangle = \langle w, w \rangle = ||w||^2 = 2P(w). \qquad (11)$$

Therefore, it follows, using (10) and (11), that given $\epsilon > 0$, $||w|| \geq \epsilon$, one can choose $\delta_n > 0$ small enough such that

$$\begin{aligned}
\langle \nabla P(w), \theta - w \rangle &= \langle \nabla P(w), \theta \rangle - \langle \nabla P(w), w \rangle \\
&\leq ||\nabla P(w)||\ \frac{2G\delta_n}{m} - 2P(w) \leq -P(w).
\end{aligned}$$

Consequently,

$$\frac{d}{dt} P\left(w(t)\right) \leq -P\left(w(t)\right),$$

so that

$$P\left(w(t)\right) \leq P\left(w(0)\right) e^{-t}.$$

This implies that $P(w(t))$ goes to zero at exponential rate and the set D^1 is a global attractor for the DI (6). Hence, the time average regret B_n and its corresponding regret C_n will then approach D^1. This completes the proof.

Theorem 2. *If all agents follow the proposed procedure, the empirical distribution of joint play of all agents $z_n(s)$ converges almost surely as $t \to \infty$ to the set of correlated equilibria in the action space, for finite payoffs.*

Proof. The proof follows from how the "regret" measure is defined. Recall that

$$\begin{aligned}[C(z_n)]_{j,k} &= \sum\nolimits_{\ell \in \mathcal{L}} z_n(j, \ell_n) \left(U(k, \ell_n) - U(j, \ell_n)\right) \\ &= \sum\nolimits_{s_n \in S : i_n = j} z_n(s_n) \left(U(k, \ell_n) - U(s_n)\right),\end{aligned}$$

where $s_n = (i_n, \ell_n)$ is the joint play made at stage n. On any convergent subsequence $\lim\limits_{n \to \infty} z_n \to \Pi$, we get

$$\lim_{n \to \infty} [C(z_n)]_{j,k} = \sum\nolimits_{s_n \in S : i_n = j} \Pi(s_n) \left(U(k, \ell_n) - U(s_n)\right) \leq 0.$$

Next, comparing with the definition of CE as in (2) completes the proof.

4 Evaluation

In this section, we evaluate the performance of our proposed algorithm using a well-known multiagent Prisoner's Dilemma game (also known as the Tragedy of the Commons) [13]. Let's consider the game in which multiple agents ($A \geq 200$) compete for a limited common resource. Each agent has to make a binary decision – "yes" or "no" that models the agent decision of using the common resource or not, respectively. The agent that does not use the resource gets a fixed payoff. All the agents using the resource get the same payoff. Consequently, the more agents decided to use the resource, the smaller the obtainable payoff per agent; and when the number of agents sharing the resource is higher than a certain threshold, it is better for the others not to use the resource. A simple utility function reflecting this game can be expressed as follows:

$$U = \begin{cases} 1 & \text{if agent decision is "no"}, \\ 101 - \eta & \text{if agent decision is "yes"}. \end{cases}$$

with η being the number of agents making the same "yes" decision.

To evaluate the performance of our solution, we analyse the two metrics:

- Convergence speed (iterations): number of iterations to convergence. A fast convergence is preferable.
- System fairness index, which is derived as

$$J = \frac{(\sum_{a=1}^{A} x_a)^2}{A \times \sum_{a=1}^{A} x_a^2}, \tag{12}$$

where x_a is the average payoff of user a and A is the number of agents. Notes that $J = 1$ is the best fairness of the system, which guaranteeing the same payoff among the agents.

It can be seen that this game has two pure Nash equilibrium points when either 99 or 100 agents use the common resource. Any solutions that yield the average number of resource agents between 99 and 100 will be in the set of correlated equilibria. Among them, the equilibrium point when $\eta = 100$ provides the best system fairness since all agents will receive the same payoff of 1.

We compare our proposed algorithm with three other algorithms:

- CODIPAS-RL in [4]: Agents learn both the expected payoff and the strategies in order to make decisions. This is a popular state-of-the-art reinforcement learning algorithm and has been shown to be superior to the conventional RL scheme such as Q-learning.
- Regret-based RL in [3]: Agents update their play probability proportional only to the estimates of "positive regret" for not having played other options.
- Our proposed algorithm: Agents update their learning rules by considering both positive and negative regrets for not choosing other options.
- Exhaustive Search: A centralised controller with complete information of the game considers all possible associations involving all agents and assigns agents decisions in a way to maximise the system fairness. We use this algorithm as a benchmark since it leads to the highest performance in fairness.

Figures 1 and 2 show, respectively, the evolution of average number of agents using the resource (resource agents) and the system fairness index for the game with 200 agents. With the same initial probabilities, we observed that our proposed algorithm achieves the fastest convergence speed among all the reinforcement learning algorithms. Our algorithm converges to equilibrium states in a very small number of iterations (less than 150 iterations), where as it requires a longer time to converge for both CODIPAS-RL (up to 400 iterations) and

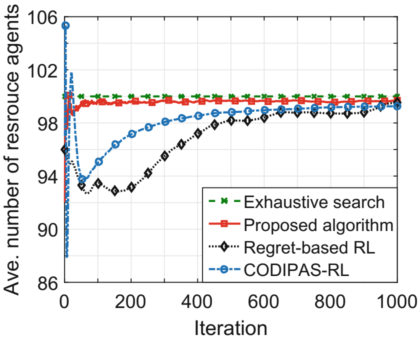

Fig. 1. Evolution of average number of resource agents by different algorithms.

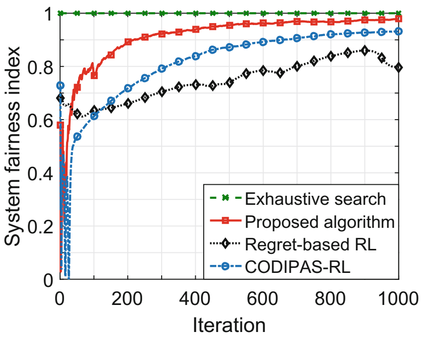

Fig. 2. Evolution of system fairness index by different algorithms.

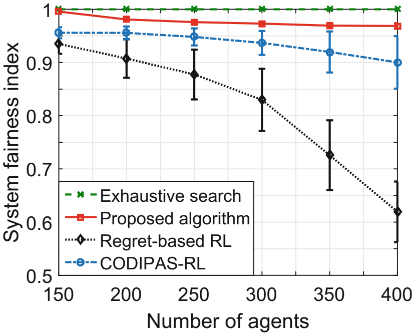

Fig. 3. Comparison of fairness between algorithms for the same number of iterations.

Regret-based RL (up to 900 iterations), especially the later. In fairness metric, our algorithm also leads to the highest system fairness index under the same number of iterations, as compared to the other RL schemes. The Regret-based RL scheme performs poorest due to its slow convergence speed.

To further study the impact of the total number of agents in the game on algorithms performance, we vary the agent number from 150 to 400 and measure the performances of all algorithms in fairness metric. The result is shown in Fig. 3. As we can see, proposed algorithm is quite robust in achieving system fairness to the change of the agent number. Increasing the total learning agents slightly reduces the system fairness index in our solution, but considerably bring down system fairness in other approaches, especially the Regret-based RL approach and when the total number of agents is very large.

5 Conclusion

We studied the problem of multiagent repeated games. We develop a fully distributed reinforcement learning procedure that takes advantage of both positive and negative regrets to speed up the learning process and improve the efficiency of the well-known regret-based reinforcement learning. Simulation results show that our solution is highly efficient with fast convergence speed and good fairness performance; and is more robust to the total number of agents in the system than other reinforcement learning algorithms. In our future research, we will study the rate of convergence of our algorithm and compare its performances on a broader set of benchmarks. As further work in this direction, a reinforcement learning framework for finding the global optimal solution in distributed multiagent system is still an open problem. Investigating the impact of irrational agents on the learning outcome is another challenging problem to consider.

Acknowledgment. This research is partially supported by the Australian Research Council Linkage Grant LP100200493.

References

1. Bhatnagar, S., Prasad, H., Prashanth, L.: Reinforcement learning. In: Bhatnagar, S., Prasad, H., Prashanth, L. (eds.) Stochastic Recursive Algorithms for Optimization, pp. 187–220. Springer, London (2013)
2. Sandholm, T.W., Crites, R.H.: Multiagent reinforcement learning in the iterated prisoner's dilemma. Biosystems **37**(1–2), 147–166 (1996)
3. Hart, S., Mas-Colell, A.: A reinforcement procedure leading to correlated equilibrium. In: Debreu, G., Neuefeind, W., Trockel, W. (eds.) Economics Essays, pp. 181–200. Springer, Berlin (2001). doi:10.1007/978-3-662-04623-4_12
4. Tembine, H.: Fully distributed learning for global optima. In: Distributed Strategic Learning for Wireless Engineers, pp. 317–359. CRC Press, UK (2012)
5. Kalathi, D., Borkar, V.S., Jain, R.: Blackwell's approachability in stackelberg stochastic games: a learning version. In: 53rd IEEE Conference on Decision and Control, pp. 4467–4472 (2014)
6. Bravo, M., Faure, M.: Reinforcement learning with restrictions on the action set. SIAM J. Control Optim. **53**(1), 287–312 (2015)
7. Borowski, H.P., Marden, J.R., Shamma, J.S.: Learning efficient correlated equilibria. In: 53rd IEEE Conference on Decision and Control, pp. 6836–6841 (2014)
8. Hart, S., Mas-Colell, A.: A simple adaptive procedure leading to correlated equilibrium. Econometrica **68**(5), 1127–1150 (2000)
9. Bowling, M.: Convergence and no-regret in multiagent learning. Adv. Neural Inf. Process. Syst. **17**, 209–216 (2005)
10. Cigler, L., Faltings, B.: Reaching correlated equilibria through multi-agent learning. In: The 10th International Conference on Autonomous Agents and Multiagent Systems, vol. 2, pp. 509–516 (2011)
11. Aumann, R.J.: Correlated equilibrium as an expression of Bayesian rationality. Econometrica **55**(1), 1 (1987)
12. Benam, M., Hofbauer, J., Sorin, S.: Stochastic approximations and differential inclusions, part II: applications. Math. OR **31**(4), 673–695 (2006)
13. Apt, K.R., Grädel, E.: A primer on strategic games. In: Apt, K.R., Grädel, E. (eds.) Lectures in Game Theory for Computer Scientists, pp. 1–37. Cambridge University Press (2011)