Automatica 113 (2020) 108615

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

A strategic learning algorithm for state-based games[☆]

Changxi Li [a], Yu Xing [b], Fenghua He [a,*], Daizhan Cheng [b]

[a] Control and Simulation Center, Harbin Institute of Technology, Harbin 150001, PR China
[b] Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, PR China

ARTICLE INFO

Article history:
Received 15 September 2018
Received in revised form 18 June 2019
Accepted 9 September 2019
Available online 20 December 2019

Keywords:
Strategic learning
State-based games
Recurrent state equilibria
Multi-agent systems

ABSTRACT

Learning algorithm design and applications of state-based games are investigated. First, a heuristic uncoupled learning algorithm, which is a two memory better reply learning rule, is proposed. Under reachability conditions it is proved that for any initial state, if all agents in the state-based game follow the proposed learning algorithm, the action state pair converges almost surely to an action invariant set of recurrent state equilibria. The design of the learning algorithm relies on global and local searches with finite memory, inertia, and randomness. Then, existence of time-efficient universal learning algorithm is studied. Finally, applications of our proposed learning algorithm are discussed, including learning pure Nash equilibrium in finite games and cooperative control with time-varying communication structure.

1. Introduction

Many systems, such as biological networks, social networks (French, 1956), and engineering systems, can be described as a collection of interacting subsystems, in which local decisions are made with local information (Marden & Shamma, 2015). It is the core mission in such systems to ensure the emergence of desirable collective behavior by designing proper local control strategies. Game-theoretical method is becoming an appealing tool in control of the above systems as it provides a modularized design architecture, i.e., interaction structures and learning algorithms can be designed separately (Marden & Shamma, 2015; Ocampo-Martinez & Quijano, 2017). Some outstanding works include: (i) consensus/synchronization of multi-agent systems (Marden & Shamma, 2015); (ii) distributed optimization (Yang & Johansson, 2010); (iii) optimization in energy (Saad, Han, Poor, et al., 2012) and transportation networks (Wang, Xiao, Wongpiromsarn, et al., 2013), just to name a few.

State-based game, an extended model in game-theoretic control, was proposed in Marden (2012), which is a simplification of stochastic games Shapley (1953). In fact, the original idea of state-based games can be traced back to Young (2004, Section 9, Conclusion). Since then state-based games have shown their strong vitality in many fields, such as achieving Pareto optimality (Marden, Young, & Pao, 2014), realizing cooperative coverage in unknown environment (Rahili & Ren, 2014), and solving distributed economic problems in smart grid (Liang, Liu, Wei, et al., 2016). Particularly, a completely uncoupled learning algorithm for general games is designed for the first time using the theory of state-based games and regular perturbed Markov chain (Young, 2009).

Compared with traditional game-theoretical framework, state-based games provide an additional degree of freedom, which is called *state*, to help coordinate group behavior. The underlying "state" has a variety of interpretations ranging from a dummy agent (Marden, 2012) or external environment (Young, 2004) to real agents with unknown dynamics or dynamics for equilibrium selection (Marden, 2017; Pradelski & Young, 2012). Because this additional degree of freedom is provided to help coordinate system level behavior, state-based game is extremely useful in game-theoretic control.

One of the core challenges in applying state-based game model to game-theoretic control is how to design a strategic learning algorithm which can converge to the equilibria of the games. Although (Marden, 2012) proposed a finite memory learning algorithm for state-based potential games, to our best knowledge, there is no strategic learning algorithm for general state-based games. The purpose of this paper is to design a heuristic algorithm for general state-based games and to discuss the applications of state-based games. In this paper, all agents are supposed to improve their one-shot payoff. The equilibrium in state-based games is called recurrent state equilibrium.

The first contribution of this paper is that a *heuristic* learning algorithm for general state-based games is developed. The proposed algorithm is a two memory better reply learning rule. The

[☆] This work is supported partly by the National Natural Science Foundation of China (NSFC) under Grants 61473099, 61773371, 61733018 and 61333001. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Rong Su under the direction of Editor Christos G. Cassandras.

[*] Corresponding author.
E-mail addresses: changxi1989@163.com (C. Li), yxing@amss.ac.cn (Y. Xing), hefenghua@hit.edu.cn (F. He), dcheng@iss.ac.cn (D. Cheng).

https://doi.org/10.1016/j.automatica.2019.108615
0005-1098/© 2019 Elsevier Ltd. All rights reserved.

two-memory strategy is used to test whether the recurrent state equilibrium is obtained, while the better-reply rule makes each agent improve its one-shot payoff. The design of the algorithm relies on global and local searches depending on two-memory information, inertia, and randomness, and its insight is illustrated intuitively. Under the reachable condition, it is proved that the algorithm converges almost surely to a recurrent state equilibrium of state-based games.

The second contribution is that several applications of the designed learning algorithm are discussed: (i) When the model is reduced to normal game (i.e., there is no state), our proposed algorithm still works. A numerical case study is provided to demonstrate the validity. (ii) An example is presented to solve cooperative control problem of multi-agents with time-varying communication structure by designing proper utility functions.

The rest of this paper is organized as follows: Section 2 provides some preliminaries, including the formal definition of state-based games, recurrent state equilibrium, and the theory of learning in state-based games. Section 3 focuses on the design of a learning algorithm for general state-based games. Section 4 considers applications of state-based game. A brief conclusion is given in Section 5. Convergence of our proposed learning algorithm is proved in Appendix A.

2. Preliminaries

2.1. State-based games

Definition 1 (*Marden, 2012*)**.** A finite state-based game is a quintuple $\mathcal{G} = \{N, \{A_i\}_{i\in N}, \{c_i\}_{i\in N}, X, P\}$ where

(1) $N = \{1, 2, \ldots, n\}$ is the set of agents;
(2) $A_i = \{1, 2, \ldots, k_i\}$ is the set of actions of agent i;
(3) $c_i : A \times X \to \mathbb{R}$ is agent i's payoff function, where $A = \prod_{i=1}^{n} A_i$ is the action profile set, and $\prod$ is the Cartesian product;
(4) $X = \{1, 2, \ldots, m\}$ is the underlying finite state set;
(5) $P : A \times X \to \Delta(X)$ is the Markovian state transition function, where $\Delta(X)$ denotes the set of probability distributions over the finite state space X.

Let $P(a; x, y)$ denote the state transition probability from state $x \in X$ to state $y \in X$ under the action $a \in A$. Denote $P(a; \cdot, \cdot)$ the probability transition matrix of a joint action $a \in A$. Obviously, a Markov chain is defined by $P(a; \cdot, \cdot)$ with state pace X.

When a state-based game is played repeatedly, a sequence of states

$$x(0), x(1), \ldots, x(t), \ldots$$

and a sequence of joint actions

$$a(0), a(1), \ldots, a(t), \ldots$$

are generated. $[a(t), x(t)] \in A \times X$ is referred to the action state pair at time t. We give a rough description on how the action state pair evolves. The sequence of action profiles is generated from some specified decision algorithm. Suppose the current state is $x(t)$, and the action taken by all agents at time t is $a(t)$, then $x(t+1)$ is generated by the state transition function $P(a(t); x(t), \cdot)$, i.e., the ensuing state is selected randomly according to the probability distribution $P(a(t); x(t), \cdot)$. The dynamics of state-based games can be described as in Fig. 1, where '$\vDash$' signifies that the ensuing state $x(k+1)$ is selected according to the probability distribution $P(a(k); x(k), \cdot)$.

Denote by $X(a|x) \subseteq X$ the set of reachable states starting from initial state x driven by an invariant action a. That is to say, a state $y \in X(a|x)$ if and only if there exists a time $t_y > 0$ such that

$$\mathbf{Pr}[x(t_y) = y] > 0,$$

conditioned on the events $x(0) = x$ and $x(k+1) \vDash P(a; x(k), \cdot)$ for all $k \in \{0, 1, \ldots, t_y - 1\}$. The transition process can be illustrated as

$$x \xrightarrow{a} x(1) \xrightarrow{a} \overset{a}{\cdots} \xrightarrow{a} x(t_y - 1) \xrightarrow{a} x(t_y) = y.$$

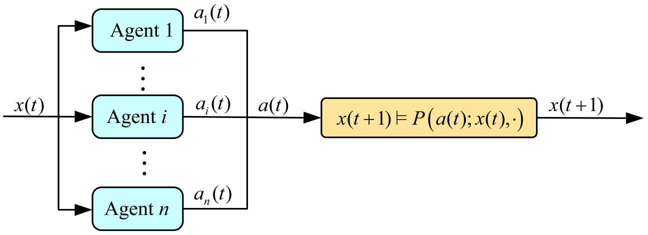


Fig. 1. Dynamics of state-based games.

Table 1
Payoff Bi-Matrix for $x = 1$ of Example 4 (coordination game).

Agent 1	Agent 2	
	1	2
1	(4, 4)	(1, 3)
2	(3, 1)	(2, 2)

Table 2
Payoff Bi-Matrix for $x = 2$ of Example 4 (prisoner's dilemma game).

Agent 1	Agent 2	
	1	2
1	(2, 2)	(0, 3)
2	(3, 0)	(1, 1)

As a generalization of Nash equilibrium, the equilibrium in state-based games is called *recurrent state equilibrium* (RSE).

Definition 2 (*Marden, 2012 Recurrent State Equilibrium*)**.** Consider a state-based game $\mathcal{G}$. The action state pair $[a^*, x^*]$ is a recurrent state equilibrium with respect to the state transition process P if the following two conditions are satisfied:

(1) The state x^* satisfies $x^* \in X(a^*|x)$ for every state $x \in X(a^*|x^*)$;
(2) For each agent $i \in N$ and every state $x \in X(a^*|x^*)$,

$$c_i(a_i^*, a_{-i}^*, x) \geqslant c_i(a_i, a_{-i}^*, x), \;\; \forall a_i \in A_i.$$

The first condition means that if the action state pair $[a^*, x^*]$ is a recurrent state equilibrium, then $X(a^*|x^*)$ is a recurrent class of the Markov chain $P(a^*; \cdot, \cdot)$ starting from the initial state x^*. The second condition implies that a^* is a pure Nash equilibrium of state invariant game $G_x := \{N, \{A_i\}_{i\in N}, \{c_i(\cdot, x)\}_{i\in N}\}$ for every state $x \in X(a^*|x^*)$.

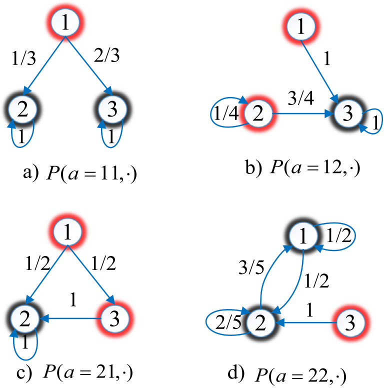
Remark 3. As the state-based game model has probabilistic transition of the states, it is obvious that evaluating long term cost can provide more accurate estimation. But in some situations agents do not know the Markovian state transition function P. Then it is impossible to calculate the long term expected cost for each agent.

Example 4. Consider the following state-based game with $N = \{1, 2\}$, $A_1 = A_2 = \{1, 2\}$, $X = \{1, 2, 3\}$. The game G_x is a coordination game, prisoner's dilemma game, and matching pennies game when $x = 1, 2$, and 3, respectively. The payoff matrices are shown in Tables 1–3. The state transition process is shown in Fig. 2.

One can verify that the recurrent states of Markov chain $P(a = 22, \cdot)$ are $x = 1, x = 2$, and $a = 22$ is a pure Nash equilibrium

Table 3
Payoff Bi-Matrix for $x = 3$ of Example 4 (matching pennies game).

Agent 1	Agent 2	
	1	2
1	$(-1,\ 1)$	$(1,\ -1)$
2	$(1,\ -1)$	$(-1,\ 1)$



a) $P(a = 11, \cdot)$ b) $P(a = 12, \cdot)$ c) $P(a = 21, \cdot)$ d) $P(a = 22, \cdot)$

Fig. 2. State transition diagram of Example 4.

when $x = 1, 2$. Therefore, action state pairs $[a = 22, x = 1]$ and $[a = 22, x = 2]$ both are the recurrent state equilibria of Example 4. Although $a = 11$ is the pure Nash equilibrium of $\mathcal{G}_1$, $x = 1$ is a transient state of Markov chain $P(a = 11, \cdot)$. So $[a = 11, x = 1]$ is not a recurrent state equilibrium.

2.2. Learning in state-based games

Consider a repeated state-based game. The observed sequence of agent i at time t is $\{\{a(\tau), x(\tau)\}_{\tau=0,\ldots,t-1}, x(t)\}$. Let $O_i(t)$ denote the obtained/available information of agent i at time t, that is,

$$O_i(t) := \Big\{ \{a(\tau), x(\tau)\}_{\tau=0,1,\ldots,t-1}, x(t) \Big\}.$$

Generally speaking, the action updating mechanism of agent i can be described by a response function f_i (Jordan, 1993),

$$f_i : O_i(t) \to \Delta(A_i),$$

where f_i is a function which maps agent i's available information $O_i(t)$ to a probability distribution over i's own actions A_i. Agent i selects the action $a_i(t+1) \in A_i$ according to this probability distribution at time $t+1$.

According to the available information used in making decisions, the most common learning algorithms can be categorized as *uncoupled* learning algorithms and *completely uncoupled* learning algorithms, whose definitions are shown as follows.

Definition 5 (*Talebi, 2013*)**.** A learning algorithm is called

(i) *uncoupled* if the available information of agent i used for decision-making is the payoff structure of himself and history sequence of the play, i.e.,

$$O_i(t) = \Big\{ \{a(\tau), x(\tau)\}_{\tau=0,1,\ldots,t-1}, x(t);\ c_i(a, x) \Big\}.$$

(ii) *completely uncoupled* if the available information of agent i used for decision-making is his own past realized payoffs and actions, i.e.,

$$O_i(t) = \Big\{ \{a_i(\tau), x(\tau), c_i(a(\tau), x(\tau))\}_{\tau=0,1,\ldots,t-1}, x(t) \Big\}.$$

The paper focuses on designing a *natural* and *effective* strategic learning algorithm which converges to a recurrent state equilibrium. By *natural* we require the algorithm being uncoupled or completely uncoupled. By *effective* we mean that the designed algorithm should converge to the equilibrium heuristically, not be trapped in an adjustment cycle, and not be predicted easily by each agent's opponents.

3. A two-memory better reply learning rule

3.1. Available information

Consider a repeated state-based game. Each agent seeks to maximize one-shot payoff. Agent i knows his own payoff function, but he does not know his opponents' ones. He can observe the current state x and his opponents' actions $a_{-i} \in A_{-i} := \prod_{j \neq i} A_j$, but the agent does not know the structure of the Markovian state transition function P. Each agent can recall the past 2-period information at each time. Denote by $\xi_i(t)$ the information used to make decision for agent i at time $t \geq 2$

$$\xi_i(t) := \big\{a(t-2), a(t-1), x(t);\ c_i(a, x)\big\}.$$

Then the response function f_i of agent i has the following form

$$p_i(t) = f_i\big(\xi_i(t)\big) \in \Delta(A_i).$$

For any action state pair $[a, x] \in A \times X$, agent i's *strict better reply set* is defined as

$$B_i(a; x) := \big\{a_i' \in A_i :\ c_i(a_i', a_{-i}, x) > c_i(a, x)\big\}.$$

For simplicity, let $B_i(t) := B_i(a(t-1); x(t)),\ \forall t \geq 1$.

3.2. The flow of the two-memory better reply learning algorithm

Suppose the information of the past two periods at time $t \geq 2$ is $[a(t-2), x(t-1)] \times [a(t-1), x(t)] \in (A \times X) \times (A \times X)$. Denote by $p_i^{a_i}(t)$ the probability that agent i selects $a_i \in A_i$ at time t. The learning algorithm is described as follows:

(i) Check whether $a(t-2) = a(t-1)$ or not at time t.

(ii) If $a(t-2) = a(t-1)$. Then each agent calculates $B_i(t)$ and checks whether $B_i(t) = \emptyset$ or not.

- If $B_i(t) = \emptyset$, then agent i plays $a_i(t-1)$ at next moment.
- If $B_i(t) \neq \emptyset$, then agent i selects an action according to the following probability distribution.

$$\begin{cases} p_i^{a_i(t-1)}(t) = \epsilon_i, \\ p_i^{a_i}(t) = \frac{1-\epsilon_i}{|B_i(t)|}, \forall a_i \in B_i(t), \end{cases} \tag{1}$$

where $\epsilon_i \in (0, 1)$ is the inertia of agent i.

(iii) If $a(t-2) \neq a(t-1)$, then all agents take actions simultaneously according to the following probability distributions.

$$\begin{cases} p_i^{a_i(t-1)}(t) = \epsilon_i, \\ p_i^{a_i}(t) = \frac{1-\epsilon_i}{|A_i|-1}, \forall a_i \in A_i \setminus \{a_i(t-1)\}. \end{cases} \tag{2}$$

Remark 6. The proposed learning algorithm is a two-memory, stochastic learning algorithm with inertia ϵ_i for agent i. It is a combination of *testing*, *searching*, and *lock-in*. Since the learning algorithm is a two-memory one, every agent can observe his opponents' actions. So each agents can tell whether $a(t-2) = a(t-1)$ or not. This is *testing*. The searching process consists of *local search* and *global search*. If $a(t-2) \neq a(t-1)$, then all agents take actions simultaneously according to their probability distributions with full support. This is a *global stochastic search*, both for agents and actions. If $a(t-2) = a(t-1)$ and $B_i(t) \neq \emptyset$, then agent i will take actions from $B_i(t)$. This is a *local random search*. If $a(t-2) = a(t-1)$ and $[a(t-2), x(t-2)]$ is an RSE, all agents will repeat their actions forever, which is called *lock-in*.

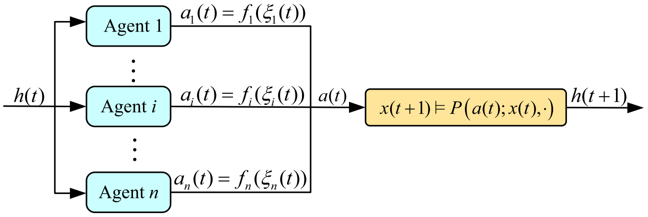


Fig. 3. Flow of the two-memory better reply rule.

Denote by $h(t) := \{a(t-2), a(t-1), x(t)\}$ the past two plays, $t > 2$. Then the flow of the two-memory better reply learning algorithm can be described as in Fig. 3.

3.3. Convergence of the proposed learning algorithm

Consider a state-based game $\mathcal{G} = \big\{N, \{A_i\}, \{c_i\}, X, P\big\}$. Let

$$\bar{P}(\cdot,\cdot) := \frac{1}{|A|}\sum_{a\in A} P(a;\cdot,\cdot),$$

and we know that $\bar{P}(\cdot,\cdot) \in \mathbb{R}^{|X|\times|X|}$ is row stochastic. Then a Markov chain is defined by $\bar{P}$ with X as its state space. Suppose $\mathcal{G}$ has at least one RSE, and let

$$A^* = \{a \in A |\ \exists x \in X, \text{ s.t. } [a,x] \text{ is a RSE}\}.$$

For $a \in A^*$, denote

$$X(a) := \{x \in X : \exists x^* \in X(a|x), \text{ s.t. } [a, x^*] \text{ is an RSE}\}.$$

The set $X(a)$, $\forall a \in A^*$ contains all states from which the algorithm can reach an RSE class of action a with positive probability by only adopting the same action a. Let $X^* := \bigcup_{a\in A^*} X(a) \subseteq X$.

Assumption 7 (*Reachability Condition*). Consider a state-based game. Suppose that either $X = X^*$, or $X \neq X^*$ and the following assumptions hold:

(i) For every recurrent class $\bar{R}$ of $\bar{P}$, there exists an action $a^* \in A$ and a state $x^* \in \bar{R}$ such that $[a^*, x^*]$ is an RSE.

(ii) $P(a; x, x) > 0$ for all $a \in A$ and $x \in X \setminus X^*$.

Theorem 8. *Consider a state-based game $\mathcal{G} = \{N, \{A_i\}, \{c_i\}, X, P\}$, where the recurrent state equilibria exist. Suppose that Assumption 7 is satisfied. Then for any initial state $x_0 \in X$, if all agents play the game $\mathcal{G}$ by the proposed two memory better reply learning algorithm, the action state pair converges almost surely to an action invariant set of recurrent state equilibria.*

Conditions in Assumption 7 guarantee that under the action of the proposed learning algorithm, there exists a positive probability "path" which leads any initial action state pair to an RSE. The proof of Theorem 8 is presented in Appendix B.

Remark 9. It is worth noting that under the proposed learning rule action state pair can converge to an action invariant set of recurrent state equilibria. Denote by $[a^*, x^*]$ the converged action state pair. The state can move around inside the recurrent class $X(a^*|x^*)$ of Markov chain $P(a^*;\cdot,\cdot)$, which causes the one stage payoff of each player to change. But most of all, the action profile, which is optimal for all the states in $X(a^*|x^*)$, does not change.

The following example shows that the assumption (ii) of Theorem 8 avoids the situation where some desired actions cannot be selected according to the learning algorithm.

Table 4
Payoff Bi-Matrix for $x = 1$ of Example 10.

Agent 1	Agent 2	
	C	D
C	(5, 4)	(2, 3)
D	(4, 2)	(3, 1)

Table 5
Payoff Bi-Matrix for $x = 2$ of Example 10.

Agent 1	Agent 2	
	C	D
C	(1, 2)	(3, 1)
D	(2, 0)	(2, 1)

Table 6
Payoff Bi-Matrix for $x = 3$ of Example 10.

Agent 1	Agent 2	
	C	D
C	(−1, 1)	(1, −1)
D	(1, −1)	(−1, 1)

Table 7
Payoff Bi-Matrix for $x = 4$ of Example 10.

Agent 1	Agent 2	
	C	D
C	(2, 2)	(2, 3)
D	(0, 3)	(3, 1)

Example 10. Consider the following state-based game with $N = \{1, 2\}$, $A_1 = A_2 = \{\text{C}, \text{D}\}$, $X = \{1, 2, 3, 4\}$, and $A = \{CC, CD, DC, DD\}$. The payoff bi-matrices are shown in Tables 4–7.

The Markovian state transition matrices are as follows:

$$P(\text{CC};\cdot,\cdot) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad P(\text{CD};\cdot,\cdot) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$P(\text{DC};\cdot,\cdot) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad P(\text{DD};\cdot,\cdot) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

It can be observed that the only RSE is $(CC, 1)$. Suppose that $x(0) = 4$, and the only possible choice of actions such that the system leaves the state 4 and reaches the state 2 is adopting CC twice. This is because $a(0)$ must be CC and $x(1) = 3$ with probability 1/2. Although $a(1)$ can be any action in A, actions CD, DC, and DD make the system return to the state 4. Therefore, $a(1)$ should be CC too, and $x(2) = 2$ with probability 1/2 on the condition that $x(1) = 3$.

However, since $B_1(\text{CC}, 2) = \{\text{D}\}$ and $B_2(\text{CC}, 2) = \emptyset$, the algorithm can only select actions from set $\{\text{CC}, \text{DC}\}$ at time $t = 2$. The choice CC makes the system stay at 2, while the latter makes $x(3) = 4$, and everything returns to the beginning. Thus, the algorithm cannot reach the RSE from the initial state $x(0) = 4$, though $\bar{P}$ is irreducible, and the assumption (i) of Theorem 8 holds.

3.4. Existence of universal time-efficient learning algorithm

One may be interested in the complexity of the proposed learning algorithm, especially the time efficiency. The time efficiency of a learning algorithm is defined as follows:

Definition 11 (*Talebi, 2013*). A learning algorithm is called *time efficient* if the time for the algorithm to converge to an equilibrium is polynomial with respect to the number of agents.

Hart and Mansour (2010) proved that there does not exist any time-efficient uncoupled learning algorithm that converges to a pure Nash equilibrium for generic normal form games where such an equilibrium exists. As state-based games contain normal form games as its special case, we can conclude that:

Proposition 12. *There does not exist any time-efficient uncoupled learning algorithms that converge to a recurrent state equilibrium for general state-based games where such an equilibrium exists.*

In fact, when it comes to state-based games, things become a bit more complicated. There is even no universal learning algorithm converging to a recurrent state equilibrium.

Remark 13. If for all Markov chain $P(a;\cdot,\cdot)$, $\forall a \in A$, there exists a common closed set, denoted by $X^c \subseteq X$, such that, for all $[a,x] \in A \times X^c$, $[a,x]$ is not an RSE. Then there does not exist any uncoupled learning algorithm that converges to an RSE for generic state-based games even if such an equilibrium exists.

The reason why there does not exist such learning algorithms is that for a given state-based game the dynamic of the state $P(a;\cdot,\cdot)$ is pre-given, which is uncontrollable.

4. Applications

4.1. Learning pure Nash equilibrium in finite games

We present the relations between the proposed learning rule and existing works.

Corollary 14.

(1) *When the state-based model is reduced to the normal games (i.e., there is no state), our proposed two-memory better reply algorithm is similar to the two-memory learning rule proposed in Hart and Mas-Colell (2006). Therefore, our proposed two-memory better reply algorithm can be used to find pure Nash equilibria in normal games where such equilibria exist.*

(2) *Our proposed learning rule is two-memory better reply learning rule, which is different with the existing rule. For example, the gradient play is suitable for normal games with continuous action set, and our algorithm is designed for finite games. And the fictitious play (Shamma & Arslan, 2005) requires that all agents remember all previous actions at each time. As for best-response rule, it can be trapped into a best reply cycle, as shown in Fig. 5.*

(3) *Marden (2012) proposed a finite memory better reply learning rule for state-based potential games. Marden (2012) proved that a one memory can ensure that the proposed learning rule converges almost surely to an action invariant set of recurrent state equilibria in state-based potential games. Our results show that the memory is at least two to converge almost surely to an action invariant set of recurrent state equilibria for general state-based games.*

We present an example to illustrate the effectiveness of the proposed 2-memory learning rule for normal form games.

Example 15. Consider a 3-player game constructed by Hart and Mas-Colell (2006), whose payoff matrix is shown in Fig. 4. Each player has three actions α, β, and γ. One can find that there is a cycle using traditional 1-memory adjustment rule, such as better response rule. Fig. 5 shows that how cycle is formed. Once the process enters in this cycle, there is no possibility to escape from it using 1-memory adjustment rule.

	α	β	γ	α	β	γ	α	β	γ
α	0,0,0	0,4,4	2,1,2	4,0,4	4,4,0	3,1,3	2,2,1	3,3,1	0,0,0
β	4,4,0	4,0,4	3,1,3	0,4,4	0,0,0	2,1,2	3,3,1	2,2,1	0,0,0
γ	1,2,2	1,3,3	0,0,0	1,3,3	1,2,2	0,0,0	0,0,0	0,0,0	6,6,6
		α			β			γ	

Fig. 4. S. Hart game (Hart & Mas-Colell, 2006).

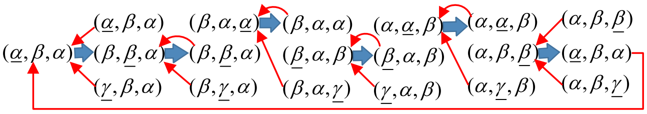

$(\underline{\alpha},\beta,\alpha)$
$(\beta,\alpha,\underline{\alpha})$
(β,α,α)
$(\alpha,\underline{\alpha},\beta)$
$(\alpha,\underline{\alpha},\beta)$
$(\alpha,\beta,\underline{\beta})$
$(\underline{\alpha},\beta,\alpha)$
$(\beta,\underline{\beta},\alpha)$
$(\beta,\underline{\beta},\alpha)$
$(\underline{\beta},\alpha,\beta)$
$(\underline{\beta},\alpha,\beta)$
$(\alpha,\beta,\underline{\beta})$
$(\underline{\alpha},\beta,\alpha)$
$(\underline{\gamma},\beta,\alpha)$
$(\beta,\underline{\gamma},\alpha)$
$(\beta,\alpha,\underline{\gamma})$
$(\underline{\gamma},\alpha,\beta)$
$(\alpha,\underline{\gamma},\beta)$
$(\alpha,\beta,\underline{\gamma})$

Fig. 5. A cycle in traditional 1-memory adjustment process.

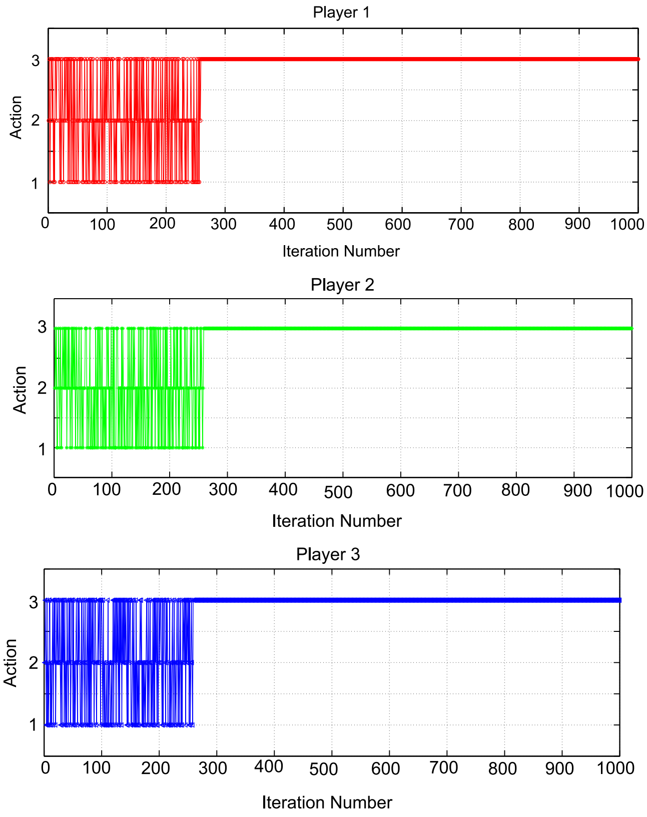


Fig. 6. Dynamics of Hart's game using the proposed learning rule with initial actions $a(0) = (2,2,2)$, $a(1) = (3,3,3)$.

However using the proposed two-memory better reply with inertia learning rule, it can converge almost surely to the pure Nash equilibrium (γ,γ,γ) in Hart's game. Denote by $1 := \alpha$, $2 := \beta$, $3 := \gamma$. The simulation results are shown in Fig. 6.

4.2. Cooperative control with time-varying communication structure

In the framework of state-based games, the underlying "state" has a variety of interpretations ranging from a dummy agent (Marden, 2012) or external environment (Young, 2004) to a real player with unknown dynamics. In other words, the state provides an additional degree of freedom for a system designer to help regulate group behavior. In the following we give an example to interpret how to realize consensus for a multi-agent system through local-information utility design using the proposed learning rule.

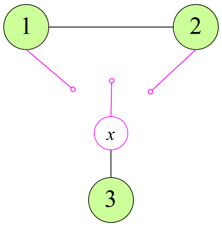


Fig. 7. MAS with time-varying communication structure.

Table 8
Markovian state transition matrices of Example 16.

$P(a,x_1)$	a							
	111	112	121	122	211	212	221	222
$p_{11}(a)$	1/3	1/4	1/2	1	1/2	0	1/3	1/3
$p_{12}(a)$	1/3	1/4	0	0	1/4	1	0	1/3
$p_{13}(a)$	1/3	1/2	1/2	0	1/4	0	2/3	1/3
$P(a,x_2)$	a							
	111	112	121	122	211	212	221	222
$p_{21}(a)$	1	0	2/3	0	0	1/2	0	1/3
$p_{22}(a)$	0	1	1/3	1/6	5/6	1/2	0	1/6
$p_{23}(a)$	0	0	0	5/6	1/6	0	1	1/2
$P(a,x_3)$	a							
	111	112	121	122	211	212	221	222
$p_{31}(a)$	1/2	1/2	1	0	1/4	0	1	1/4
$p_{32}(a)$	0	1/2	0	1/2	0	1	0	1/4
$p_{33}(a)$	1/2	0	0	1/2	3/4	0	0	1/2

Example 16. Consider a multi-agent system (MAS) with three agents $N = \{1,2,3\}$. Each agent has two actions, i.e. $A_i = \{1,2\}, i = 1,2,3$. The communication structure x, which is shown in Fig. 7, is time-varying. Define the state set as $X = \{x_1, x_2, x_3\}$, where x_1 means x connecting with agent 1, x_2 means x connecting with agent 2, and x_3 means x disconnecting. The dynamics of state x is a Markovian state transition process, which is shown in Table 8. Assume each agent can observe its neighbor's actions. The system level goal is to realize consensus at $(2,2,2)$, regardless of which state it is. To realize the system level goal, the technique is to convert this problem into a state-based games with $[(2,2,2),x]$ as its recurrent state equilibrium, $\forall x \in X$. Then using the proposed learning rule, the system will converge to the recurrent state equilibrium. The following is the detailed design procedure.

(a) State evolution process analysis:

The dynamics of the state x is a Markov chain under joint actions $a \in A := A_1 \times A_2 \times A_3$. State x_i transfers to state x_j under action a with probability $p_{ij}(a)$, $\forall i,j = 1,2,3$. According to Table 8, one can verify that the Markov chain $P(a = 222; \cdot, \cdot)$ is aperiodic and irreducible. Therefore we can design the utility function such that the joint action $(2,2,2)$ is a pure Nash equilibrium for all $x \in X$.

(b) Local-information utility design:

As each agent can only observe its neighbor's actions, the designed utility function of each agent should satisfy: (i) local information based, i.e. $c_i(a,x) = c_i(a_{N_i}, a_i, x), \forall i \in N$, where N_i is the neighbor of player i; (ii) with $(2,2,2)$ as its Nash equilibrium; and (iii) Assumptions in Theorem 8. The designed utility functions are shown in Table 9. One can verify that the designed utility functions satisfy the requirements (i), (ii) and (iii). In fact, the designed utility functions can guarantee that $[(2,2,2),x], \forall x \in X$ is an action invariant set of recurrent state equilibria of the designed state-based game, which satisfies Assumptions in Theorem 8. Here $c_i(a,x)$ is the payoff of player i when the joint action is a and the state is x.

Table 9
Designed utility function of Example 16.

$c_i(a,x_1)$	a							
	111	112	121	122	211	212	221	222
c_1	1	0	0	−1	1	1	2	3
c_2	1	1	2	2	1	1	3	3
c_3	−1	0	−1	0	1	3	1	3
$c_i(a,x_2)$	a							
	111	112	121	122	211	212	221	222
c_1	1	1	3	3	0	2	5	5
c_2	1	0	3	4	5	2	4	7
c_3	1	0	−1	2	1	0	−1	2
$c_i(a,x_3)$	a							
	111	112	121	122	211	212	221	222
c_1	1	1	0	0	−1	−1	4	4
c_2	2	2	3	3	1	1	5	5
c_3	2	3	2	3	2	3	2	3

(c) Simulation results:

The initial joint actions of all agents are $a(0) = (1,2,2)$ and $a(1) = (2,1,2)$. Denote $1 := x_1$, $2 := x_2$, $3 =: x_3$. The initial state is $x(0) = 3$. As we can see in Fig. 9, using the proposed learning rule the joint actions converge to $(2,2,2)$ after 20 steps, regardless of which state it is, as is shown in Fig. 8.

5. Conclusion

An extended model in game theory, called state-based games, is investigated in this paper. An uncoupled two memory better reply learning algorithm is proposed. We proved that under reachable conditions the proposed learning algorithm converges to a recurrent state equilibrium of a state-based games. The existence of time-efficient universal learning algorithm is also investigated. It is proved that using the learning algorithm proposed in this paper, one can realize learning pure Nash equilibrium in finite normal games and cooperative control with time-varying communication structure. Since an additional degree of freedom is provided to help coordinate group behavior, state-based game is a useful extended model in game-theoretic control.

Future works include: (i) applications of the state-based game model and the learning algorithm to engineering control problems; (ii) taking the long run average cost for evaluation in the state-based games.

Appendix A. The proposed algorithm and corresponding Markov chain

The proposed two-memory learning algorithm defines a discrete-time Markov chain $\{\omega(t), t \geq 0\}$ with finite state space $\Omega := X \times A \times X \times A \times X$, where $\omega(t) = [x(t), a(t), x(t+1), a(t+1), x(t+2)]^T$, $t \geq 0$.

Let $x^i \in X$ and $a^i \in A$ be the state and action at time i, respectively. The initial distribution of the Markov chain $\{\omega(t)\}$ is

$$\mathbf{Pr}\left\{\omega(0) = [x^0, a^0, x^1, a^1, x^2]^T\right\}$$
$$= \left(\prod_{1 \leq i \leq n} \frac{1}{|A_i|}\right)^2 p(x^0)\mathrm{P}(a^0; x^0, x^1)\mathrm{P}(a^1; x^1, x^2),$$

where $p : X \to [0,1]$ is the probability distribution of the initial state. For the sake of simplification, suppose the inertia of agent i is the same, i.e., $\epsilon = \epsilon_i$.

Consider any two states $\omega_1, \omega_2 \in \Omega$ of the Markov chain $\{\omega(t)\}$, where $\omega_1 = [x^1, a^1, x^2, a^2, x^3]^T$ and $\omega_2 = [y^1, b^1, y^2, b^2, y^3]^T$. According to the learning algorithm, the transition probability from ω_1 to ω_2 of the Markov chain $\{\omega(t)\}$ is as follows:

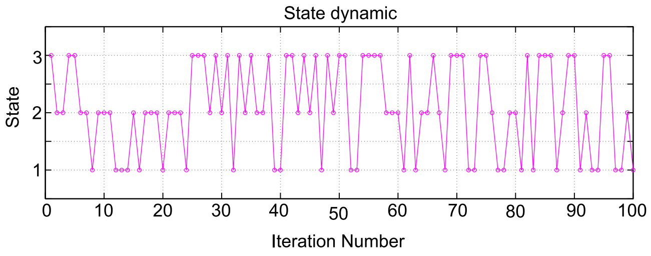


Fig. 8. Dynamics of states of Example 16.

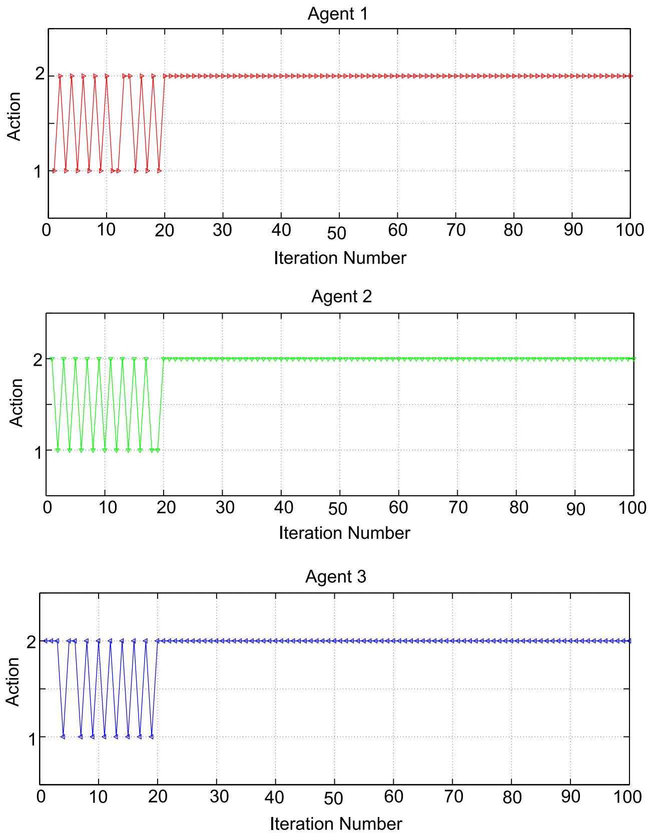


Fig. 9. Dynamics of actions of each agent in Example 16.

(1) If $[y^1,b^1,y^2]\neq[x^2,a^2,x^3]$, then

$$\mathbf{Pr}\left\{\omega(t+1)=\omega_2|\omega(t)=\omega_1\right\}=0.$$

(2) If $[y^1,b^1,y^2]=[x^2,a^2,x^3]$ and $a^1\neq a^2$, then

$$\begin{aligned}&\mathbf{Pr}\left\{\omega(t+1)=\omega_2|\omega(t)=\omega_1\right\}\\&=\epsilon^{n-|H(b^1,b^2)|}\cdot\prod_{i\in H}\frac{1-\epsilon}{|A_i|-1}\cdot\mathrm{P}(b^2;y^2,y^3),\end{aligned}$$

where $H(a,b):=\{i\in N: a_i\neq b_i\}$, $a,b\in A$.

(3) If $[y^1,b^1,y^2]=[x^2,a^2,x^3]$ and $a^1=a^2$, then

$$\begin{aligned}&\mathbf{Pr}\left\{\omega(t+1)=\omega_2|\omega(t)=\omega_1\right\}\\&=\epsilon^{n-|H(b^1,b^2)|-|N(b^1,y^2)|}\times\mathrm{P}(b^2;y^2,y^3)\\&\times\prod_{i\in H}\frac{1-\epsilon}{|B_i(b^1,y^2)|}I_{B_i(b^1,y^2)}((b^2)_i),\end{aligned}$$

where $N(a,x):=\{i\in N: B_i(a,x)=\emptyset\}$, and $I_{B_i(a,x)}(b_i)$ is an indicator function such that $I_{B_i(a,x)}(b_i)=1$ if $b_i\in B_i(a,x)$ and $I_{B_i(a,x)}(b_i)=0$ if $b_i\notin B_i(a,x)$, $a\in A, x\in X, b_i\in A_i$.

Appendix B. Proof of Theorem 8

Denote $D(a,x):=\{b\in A: b_i\in B_i(a,x)\cup\{a_i\}, i\in N\}$ as the collection of action vectors whose ith entry is a_i or a strict better-reply action for (a,x). From the definition, we know that $\{a\}\subseteq D(a,x)\subseteq A$ for any $a\in A$ and $x\in X$.

Lemma 17. *Consider a state-based game, where the RSE exists. For any fixed initial value $x(0)=x^0$ and fixed action–state pairs (a^0,x^1), (a^1,x^2) of the learning algorithm, if there exists a positive integer $K\geq 2$ and a sequence of action–state pairs $\{(a^i,x^{i+1}),\ 2\leq i\leq K\}$, where $a^i\in A$, $x^{i+1}\in X$, $2\leq i\leq K$, such that*

(i) *$P(a^2;x^2,x^3)P(a^3;x^3,x^4)\cdots P(a^K;x^K,x^{K+1})>0$;*

(ii) *if $a^{k-1}=a^k$ for some integer $k\in[1,K)$, then $a^{k+1}\in D(a^k,x^{k+1})$;*

(iii) *(a^K,x^{K+1}) is an RSE,*

then the algorithm converges to some RSE almost surely, by which we mean that $\mathrm{P}\{\tau<\infty\}=1$, where $\tau:=\min\{t\geq 2: (a^t,x^{(t+1)})$ is an RSE$\}$, and, at the same time, that $a^{(\tau+t)}=a^\tau$, $x^{(\tau+t)}\in X(a^\tau|x^{(\tau+1)})$ for $t\geq 1$.

Proof. For convenience, let

$$\omega(t):=[x^t,a^t,x^{t+1},a^{t+1},x^{t+2}]^T,\ \forall t\geq 0,$$

unless elsewhere stated. The assumptions imply that, for any fixed initial state $\omega(0)=[x^0,a^0,x^1,a^1,x^2]^T$,

$$\mathbf{Pr}\{\omega(K-1)|\omega(0)\}>0.$$

From the transition probability of $\{\omega(t)\}$ and that (a^K,x^{K+1}) is an RSE, it follows that

$$\begin{aligned}&\mathbf{Pr}\{\omega(K+1)=[x^{K+1},a^K,x^{K+2},a^K,x^{K+3}]^T\\&\quad|\omega(K-1)=[x^{K-1},a^{K-1},x^K,a^K,x^{K+1}]^T\}>0,\end{aligned}$$

where $x^{K+2},x^{K+3}\in X(a^K|x^{K+1})$.

Thus,

$$\mathbf{Pr}\{\omega(K+1)=[x^{K+1},a^K,x^{K+2},a^K,x^{K+3}]^T|\omega(0)\}>0.$$

Therefore, the algorithm can reach an RSE from any state $\omega(0)\in\Omega$ with positive probability. □

Lemma 18. *Consider a state-based game. Suppose that the following assumptions hold:*

(i) *$\bar{P}$ is irreducible;*

(ii) *there exists an action $a^*\in A$ and a state $x^*\in X$ such that (a^*,x^*) is an RSE;*

(iii) *$P(a;x,x)>0$ for all $a\in A$ and $x\in X$.*

Then for any initial state $x\in X$, the algorithm converges to some RSE class a.s.

Proof. It suffices to validate the conditions in Lemma 17 hold.

(i) For any fixed initial state $[x^0,a^0,x^1,a^1,x^2]$, if $a^0\neq a^1$, and (a^1,x^2) is an RSE, then the desired sequence of action–state pairs is obtained when we let $a^2=a^1$. If (a^1,x^2) is not an RSE but $x^2\in X(a^*|x^*)$, then let $a^2=a^*$, and the desired sequence is obtained too.

Now assume $a^0\neq a^1$, that (a^1,x^2) is not an RSE, and $x^2\notin X(a^*|x^*)$. From assumption (i), it follows that, for $x^2\in X$, there exists a positive integer $K_1\geq 3$ such that

$$\bar{P}(x^2,x^3)\bar{P}(x^3,x^4)\cdots\bar{P}(x^{K_1-1},x^{K_1})>0,$$

where $x^i \neq x^*$, $2 \leq i < K_1$, and $x^{K_1} = x^*$. The definition of $\bar{P}$ implies that there exists a sequence of action–state pairs $\{(a^i, x^{i+1}),\ 2 \leq i < K_1\}$ such that

$$P(a^2; x^2, x^3)P(a^3; x^3, x^4) \cdots P(a^{K_1-1}; x^{K_1-1}, x^*) > 0,$$

where $x^i \neq x^*$, $2 \leq i < K_1$. Let $a^{K_1} = a^*$.

Without loss of generality, suppose that (a^i, x^{i+1}) is not an RSE for all $2 \leq i < K_1$. Otherwise let $\tilde{K}_1 := \min\{2 \leq i < K_1 : (a^i, x^{i+1})$ is an RSE$\}$ and consider the sequence $\{(a^i, x^{i+1}),\ 0 \leq i \leq \tilde{K}_1\}$.

Suppose that there exists some integer $k \in [1, K_1)$ such that $a^{k-1} = a^k$ but $a^{k+1} \notin D(a^k, x^{k+1})$. Denote $\hat{k} := 1 + \max\{t \in [0, k-1) : a^t \neq a^{k-1}\}$. The assumption $a^0 \neq a^1$ implies that $\hat{k} \geq 1$. Insert an action $\tilde{a}^i \neq a^i$ between a^i and a^{i+1}, $\hat{k} \leq i < k$. In fact, $\tilde{a}^i$, $\hat{k} \leq i < k$, can be the same action vector. Assumption (iii) ensures that

$$P(a^{\hat{k}}; x^{\hat{k}}, x^{\hat{k}+1})P(\tilde{a}^{\hat{k}}; x^{\hat{k}+1}, x^{\hat{k}+1})P(a^{\hat{k}+1}; x^{\hat{k}+1}, x^{\hat{k}+2}) \cdots$$
$$P(a^{k-1}; x^{k-1}, x^k)P(\tilde{a}^{k-1}; x^k, x^k)P(a^k; x^k, x^{k+1}) > 0.$$

The condition (ii) in Lemma 17 is satisfied for this new sequence of action–state pairs, and the desired sequence is obtained in this way.

(ii) If $a^0 = a^1$, and (a^1, x^2) is an RSE, then let $a^2 = a^1$ and $x^3 \in X(a^1|x^2)$.

(iii) If $a^0 = a^1$, but (a^1, x^2) is not an RSE, then, according to the learning rule, one can choose $a^2 \neq a^1$. By applying the argument above to $(x^1, a^1, x^2, a^2, x^3)$, we can obtain the desired sequence of action–state pairs. □

Lemma 19. *Consider a state-based game. Suppose that $X = X^*$. Then for any initial state $x \in X$, the algorithm converges to some RSE class a.s.*

Proof. (i) For any fixed initial state $[x^0, a^0, x^1, a^1, x^2]$, if $a^0 \neq a^1$, and (a^1, x^2) is an RSE, then the desired sequence of action–state pairs is obtained when we let $a^2 = a^1$. Denote by $[a^*, x^*]$ an RSE. If $x^* \in X(a^*|x^2)$, then let $a^\tau = a^*$, $\tau \geq 2$, and the desired sequence is obtained too.

(ii) If $a^0 \neq a^1$, and (a^1, x_2) is not an RSE, and that $x^2 \notin X(a^*|x^*)$. As $X = X^*$, then for any $x \in X$, there exists an action $b^* \in A^*$, such that $[b^*, y^*]$ is an RSE, and $y^* \in X(b^*|x_2)$. Then let $a^\tau = b^*$, $\tau \geq 2$, and the desired sequence is obtained too.

(iii) If $a^0 = a^1$, and (a^1, x^2) is an RSE, then let $a^2 = a^1$ and $x^3 \in X(a^1|x^2)$.

(iv) If $a^0 = a^1$, but (a^1, x^2) is not an RSE, then, according to the learning rule, one can choose $a^2 \neq a^1$. By applying the argument above to $(x^1, a^1, x^2, a^2, x^3)$, we can obtain the desired sequence of action–state pairs. □

Lemma 20. *Consider a state-based game. Suppose that $X \neq X^*$ and the following assumptions hold:*

(i) for every recurrent class $\bar{R}$ of $\bar{P}$, there exists an action $a^* \in A$ and a state $x^* \in \bar{R}$ such that (a^*, x^*) is an RSE;

(ii) $P(a; x, x) > 0$ for all $a \in A$ and $x \in X$.

Then for any initial state $x \in X$, the algorithm converges to some RSE class a.s.

Proof. From the proof of Lemma 18, it suffices to show that the conditions in Lemma 17 still hold when $a^0 \neq a^1$, and x^2 is a transient state of $\bar{P}$. If there exists an action $a^* \in A$ such that (a^*, x^2) is an RSE, then let $a^2 = a^*$ and the desired sequence is obtained. Otherwise, since x^2 is transient for $\bar{P}$, we know that there exists a positive integer $K_1 \geq 3$ and a recurrent state of $\bar{P}$, $\tilde{x}$, such that

$$\bar{P}(x^2, x^3)\bar{P}(x^3, x^4) \cdots \bar{P}(x^{K_1-1}, x^{K_1}) > 0,$$

where $x^i \neq \tilde{x}$, $2 \leq i <_{\bar{}} K_1$; $x^{K_1} = \tilde{x}$; $(\tilde{a}, \tilde{x})$ is an RSE for some $\tilde{a} \in A$. The definition of $\bar{P}$ implies that there exists a sequence of action–state pairs $\{(a^i, x^{i+1}),\ 2 \leq i < K_1\}$ such that

$$P(a^2; x^2, x^3)P(a^3; x^3, x^4) \cdots P(a^{K_1-1}; x^{K_1-1}, \tilde{x}) > 0,$$

where $x^i \neq \tilde{x}$, $2 \leq i < K_1$. Let $a^{K_1} = \tilde{a}$.

We can obtain the desired sequence by applying the same argument in Lemma 18. □

Summarizing Lemmas 19 and 20, we conclude that for any initial state $x \in X$, the algorithm converges to some RSE class a.s. when the reachable condition is satisfied.

References

French, J. R. P. (1956). A formal theory of social power. *Psychological Review*, *63*(3), 181–194.

Hart, S., & Mansour, Y. (2010). How long to equilibrium? The communication complexity of uncoupled equilibrium procedures. *Games and Economic Behavior*, *69*(1), 107–126.

Hart, S., & Mas-Colell, A. (2006). Stochastic uncoupled dynamics and Nash equilibrium. *Games and Economic Behavior*, *57*(2), 286–303.

Jordan, J. S. (1993). Three problems in learning mixed-strategy Nash equilibria. *Games and Economic Behavior*, *5*(3), 368–386.

Liang, Y., Liu, F., Wei, W., et al. (2016). State-based potential game approach for distributed economic dispatch problem in smart grid. In *the proceedings of IEEE power and energy society general meeting*, (pp. 1–5).

Marden, J. R. (2012). State based potential games. *Automatica*, *48*(12), 3075–3088.

Marden, J. R. (2017). Selecting efficient correlated equilibria through distributed learning. *Games and Economic Behavior*, *106*, 114–133.

Marden, J. R., & Shamma, J. S. (2015). Game theory and distributed control. *Handbook of Game Theory with Economic Applications*, *4*, 861–899.

Marden, J. R., Young, H. P., & Pao, L. Y. (2014). Achieving pareto optimality through distributed learning. *SIAM Journal on Control and Optimization*, *52*(5), 2753–2770.

Ocampo-Martinez, C., & Quijano, N. (2017). Game-theoretical methods in control of engineering systems: an introduction to the special issue. *IEEE Control Systems*, *37*(1), 30–32.

Pradelski, B. S. R., & Young, H. P. (2012). Learning efficient Nash equilibria in distributed systems. *Games and Economic Behavior*, *75*(2), 882–897.

Rahili, S., & Ren, W. (2014). Game theory control solution for sensor coverage problem in unknown environment. In *the proceedings of 53rd IEEE conference on decision and control*, (pp. 1173-1178).

Saad, W., Han, Z., Poor, H., et al. (2012). Game-theoretic methods for the smart grid: an overview of microgrid systems, demand-side management, and smart grid communications. *IEEE Signal Processing Magazine*, *29*, 86–105.

Shamma, J. S., & Arslan, G. (2005). Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria. *IEEE Transactions on Automatic Control*, *50*(3), 312–327.

Shapley, L. S. (1953). Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America*, *39*(10), 1095–1100.

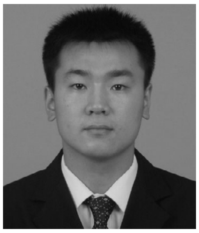
Talebi, M. S. (2013). Uncoupled learning rules for seeking equilibria in repeated plays: An overview. *Computer Science*, 1–29.

Wang, X., Xiao, N., Wongpiromsarn, T., et al. (2013). Distributed consensus in noncooperative congestion games: an application to road pricing. In *Proc. 10th IEEE Int. Conf. Contr. Aut.*, (pp. 1668–1673).

Yang, B., & Johansson, M. (2010). Distributed optimization and games: A tutorial overview. *Networked Control Systems*, *406*, 109–148.

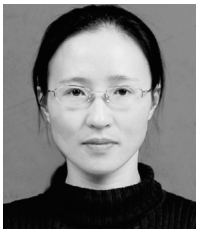
Young, H. P. (2004). *Strategic learning and its limits*. Oxford, U.K: Oxford Univ. Press.

Young, H. P. (2009). Learning by trial and error. *Games and Economic Behavior*, *65*(2), 626–643.



Changxi Li received the B.S. degree in Mathematics and Applied Mathematics and M.S. degree in Control Science and engineering from the Harbin Institute of Technology and Technology, Weihai, China, in 2013 and 2015, respectively. He is currently working toward the Ph.D. degree. His research interests include game theory and multiagent systems.

Yu Xing received his B.S. degree in Psychology from Peking University, Beijing, China, in 2014. Now he is a Ph.D. candidate in Operations Research and Control Theory at Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China. His research interests include social opinion dynamics and system identification.



Fenghua He received the Ph.D. degree in Control Science and Engineering from the Harbin Institute of Technology, Harbin, China, in 2005. She is currently a Professor with the Control and Simulation Center, Harbin Institute of Technology. Her current research interests include cooperative control, and game theory.

Daizhan Cheng (F' 06) received the Bachelor Degree from Tsinghua University, Beijing, China, the M.S. degree from the Graduate School, Chinese Academy of Sciences, Beijing, and the Ph.D. degree from Washington University, St. Louis, WA, USA, in 1970, 1981, and 1985, respectively.

He has been a Professor with the Institute of Systems Science, AMSS, Chinese Academy of Sciences, Beijing, since 1990. He is the author or co-author of 12 books, over 250 journal papers, and over 150 conference papers. His research interests include nonlinear control systems, switched systems, Hamiltonian systems, Boolean control networks, and game theory.

Dr. Cheng is an IEEE Fellow since 2006 and an IFAC Fellow since 2008. He was the Chairman of Technical Committee on Control Theory, Chinese Association of Automation from 2003 to 2010, a member of the IEEE CSS Board of Governors in 2009 and 2014, and IFAC Council Member from 2011 to 2014.

He received the Second Grade National Natural Science Award of China in 2008 and 2014, the Automatica 2008–2010 Best Theory/Methodology Paper Award, bestowed by IFAC, in 2011.