Economic Research Institute Study Paper

ERI #2004-02

THE EQUIVALENCE OF EVOLUTIONARY GAMES AND DISTRIBUTED MONTE CARLO LEARNING

by

YUYA SASAKI

Department of Economics
Utah State University
3530 Old Main Hill
Logan, UT 84322-3530

January 2004

THE EQUIVALENCE OF EVOLUTIONARY GAMES AND DISTRIBUTED MONTE CARLO LEARNING

Yuya Sasaki, Ph.D. Student

Department of Environment and Society
and
Department of Economics
Utah State University
3530 Old Main Hill
Logan, UT 84322-3530

The analyses and views reported in this paper are those of the author(s). They are not necessarily endorsed by the Department of Economics or by Utah State University.

Utah State University is committed to the policy that all persons shall have equal access to its programs and employment without regard to race, color, creed, religion, national origin, sex, age, marital status, disability, public assistance status, veteran status, or sexual orientation.

Information on other titles in this series may be obtained from: Department of Economics, Utah State University, 3530 Old Main Hill, Logan, UT 84322-3530.

myjournal manuscript No.
(will be inserted by the editor)

The equivalence of evolutionary games and distributed Monte Carlo learning

Yuya Sasaki[1]

Department of Environment and Society, Utah State University, 5215 Old Main Hill Logan, UT 84322-5215, USA; and Department of Economics, Utah State University, 3530 Old Main Hill Logan, UT 84322-3530 USA,

e-mail: `slk1r@cc.usu.edu`

Received: date / Revised version: date

Abstract This paper presents a tight relationship between evolutionary game theory and distributed intelligence models. After reviewing some existing theories of replicator dynamics and distributed Monte Carlo learning, we make formulations and proofs of the equivalence between these two models. The relationship will be revealed not only from a theoretical viewpoint, but also by experimental simulations of the models by taking a simple symmetric zero-sum game as an example. As a consequence, it will be verified that seemingly chaotic macro dynamics generated by distributed micro-decisions can be explained with theoretical models.

Key words evolutionary game – replicator dynamics – agent based models – Monte Carlo learning – recency-weighted learning

JEL-classification: C73, C63.

1 Introduction: significance of the topic

Evolutionary game theory and computational economics[1] are two of the latest fields to account for evolutionary processes of economics. When it comes to a problem of aggregate behavior, the former seems to have a limited ability to incorporate micro behaviors of each economic agent, while the latter often fails in formalizing the processes. Indeed, computational simulation practitioners empirically know that a collection of rule-based or learnable autonomous agents usually leads to an emergent outcome that partially agrees with the theoretically expected one, but the process by which distributed intelligence translates to such an outcome has been seldom understood (the difficulty of rigorous analyses is stated by Maes (1995)). However, the field also exhibits flexibilities which can be associated with existing theories of dynamics.

In this paper, we present the equivalence between the prototype theory of deterministic evolutionary games and the computational model of

[1] Computational economics varies in its sub-fields. For example, there are (i) numerical dynamics, (ii) numerical optimization (iii) heuristic optimization, and (iv) bottom-up processes by autonomous entities, etc. In this paper, we will narrow-sightedly refer to only (iv) as "computational economics."

distributed-agent Monte Carlo learning. Monte Carlo learning was selected mainly for two reasons. First, it lays a foundation for many of the other learning algorithms, especially Q-learning and Sarsa algorithms (Barto and Singh, 1990; Sutton and Barto, 1998). Second, unlike evolutionary algorithms, its simple form makes it easier to extend the algorithm to be associated with economic theory. Before discussing the main ideas, let us identify the relative roles and the historical background of evolutionary game theory, computational distributed-agent models, and learning.

Several methodologies have been developed to explain aggregate behaviors of multiple economic agents where strategic decision-making is involved. Normal-form theoretical models have played a basic role. While bearing out the theory of games von Neumann devoted himself to the development of self-reproductive machines and cellular automata (von Neumann, 1966), that later was to produce what is called artificial life or ALife (Langton, 1989). Artificial life has in turn motivated computer-based experimental scientists to model games of complex systems that generate emergent dynamics. However, few researchers have made attempts to merge these two distant fields of von Neumann's legacy [2].

While some economists were acquiring a new model of games which stem from the initial developments by Maynard Smith (1974), biologists had

[2] Some early 1990s' pioneering works (eg. Holland and Miller (1991); Arthur (1993)) have attempted to merge economic theory and autonomous intelligence models. Judd (1997) and Judd (2001) discuss the potential roles of computational economics in emerging economic theory.

started to employ computational and individual-based models (IBMs) to examine bottom-up behaviors of ecological dynamics (eg. Dewdney, 1984). Today, the contributions of IBMs are not restricted to applied problems, but include theoretical problems in population and community ecology (Haefner, 1996). On the other hand, the theoretical dynamics model of deterministic evolutionary games, or replicator dynamics (Taylor and Johnker, 1978; Schuster and Sigmund, 1983)[3], was shown by Hofbauer (1981) to be equivalent to an ecological dynamics model, namely the Lotka-Volterra equation. This sequence of events suggests the possibility of merging IBMs into the formulations of deterministic evolutionary games.

While the deterministic evolutionary games were becoming obsolete for game theorists, they certainly absorbed economists in the 1990s (Friedman (1991); and for stochastic version later by Kandori et al *et al* (1993)). Unlike the passive being of chromosomes in biological systems, economic agents exhibit active characteristics (ie. learning). It was therefore a natural course for economists to turn their attention to this dynamic aspect (eg. Roth and Erev, 1995; Dosi, 1996; Erev and Roth, 1998; Fudenberg and Levine, 1998, etc.). Computational experiments as well as psychological laboratory experiments complement theory in this aspect. One way to demonstrate how to use computational experiments in learning is to employ agent-based

[3] Extensions and analyses of the replicator dynamics, multi-agent dynamics, and other derivatives were made by Hofbauer and Sigmund (1988; 1998; 2003), Cressman (1992), Samuelson and Zhang (1992), Swinkels (1993), Weibull (1997), etc.

simulations or agent-based models (ABMs; also called agent-based computational economics or ACE), a sub-field of computational economics (for an overview, see Tesfatsion, 2002).

ABMs in economics are the analogues of IBMs in biology. In ABMs, some other forms of learnable functions (eg. Monte Carlo sampling, statistical learning, reinforcement learning, neural networks, evolutionary algorithms, etc.) are embedded in each economic agent, and the agents behave autonomously by querying their internal function for the optimal actions or strategies given the current state of the world. Hence, ABMs conduct the process of strategy optimization at a micro level. ABMs are usually employed to explain the sophisticated bottom-up processes of evolutionary economics, which would be infeasible with top-down theoretical models (Tesfatsion, 2000). While it appears that the principles of ABMs researchers and those of game theorists have diverged more than converged, the concept of the replicator dynamics roughly agrees with simple learnable ABMs, as will be shown in later sections.

In this paper we begin by briefly reviewing the basis of existing theories of deterministic evolutionary games and the algorithm of distributed Monte Carlo learning (Sec. 2). Then, our discussion moves on to formulations and proofs of the equivalence of evolutionary game theory and multi-agent Monte Carlo learning (Sec. 3). Our method does not rely on the field's convention of using the hypothetico-deductive approach, but will instead start with the somewhat farfetched connection of the two models' formulations. In Sec. 4, experimental results of ABMs will be compared to the theoretical dynamics model to graphically verify their common behavioral patterns. The end product is the theoretical and experimental verifications of the relationship between these two models. This will also enable some explanations about how micro behaviors translate to macro dynamics.

2 Background of theory and algorithm

We begin with some definitions and a description of the tools to be used for the succeeding analyses. Assume that the number of participating agents in the game world is finite and fixed. Assume also that the number of strategies or actions that these agents can take is finite and fixed. Let n denote the total number of available strategies, and let x_i denote the relative frequency of agents that take strategy i, such that $\sum_{i=0}^{n} x_i = 1, x_i \geq 0 \forall i$. Note that $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ represents the distribution vector[4] of strategy frequencies, whereas $\mathbf{x}$ in a normal-form game would represent the distribution vector of strategy probabilities. Let $\mathit{\Gamma}^n$ denote the subset of $\mathtt{R}_+^n$ defined as $\mathit{\Gamma}^n = \{\mathbf{x} \in \mathtt{R}_+^n \mid \sum_{i=1}^{n} x_i = 1,\, \mathbf{x} \geq \mathbf{0}\}$. Point $\mathbf{x}$ can move only on $\mathit{\Gamma}^n$, the strategy space. The aggregate pure states are represented by the vertices of $\mathit{\Gamma}^n$, and the aggregate mixed states by all the points off the vertices. The core rule of the game is determined by a payoff matrix $\mathbf{A}_{n \times n} = (a_{ij})$, where a_{ij} or, $\mathbf{e}_i \mathbf{A} \mathbf{e}_j$, is the payoff for taking pure strategy i when all the agents in the

[4] The vector notation used throughout this paper ignores the row/column distinctions. Thus, $\mathbf{xAx}$ means $\mathbf{x}^{\mathrm{T}}\mathbf{Ax}$.

game world would take pure strategy j. In general, the payoff for taking pure strategy i is given by $\mathbf{e}_i\mathbf{A}\mathbf{x}$. Additionally, $\mathbf{x}\mathbf{A}\mathbf{x}$ gives the average payoff to agents in the game world, since $\mathbf{x}$ is the distribution of frequencies. At an individual level, the task of strategy optimization is to choose the strategy i such that $i = \arg\max_k \mathbf{e}_k\mathbf{A}\mathbf{x}$. This strategy will surely be a pure strategy for the individual[5], meaning that the optimal strategy for an individual cannot occur in $\mathrm{int}\mathit{\Gamma}^n$. Given this, let us define the individuals' (pure) strategy set $\mathtt{P} = \{i \in \mathtt{N}^1 \mid \mathbf{e}_i \in \mathit{\Gamma}^n\}$. However, the Nash equilibrium and stable points may occur at $\mathbf{x} \in \mathrm{int}\mathit{\Gamma}^n$ in the "aggregate" level. By definition, the Nash equilibrium is the state $\mathbf{x}^*$ where

$$\mathbf{x}^*\mathbf{A}\mathbf{x}^* \geq \mathbf{x}\mathbf{A}\mathbf{x}^* \quad \forall \mathbf{x} \in \mathit{\Gamma}^n. \tag{1}$$

All these definitions are consistent with those of normal-form games, except that frequency is substituted for probabilities.

2.1 A brief review of prototype deterministic evolutionary games

It is usually feasible to analyze local behaviors of dynamics even without complete solutions, as we may identify the ω-limit (α-limit) of a dynamics if the stability (unstability) exists around the equilibrium. An important concept used in evolutionary games is the evolutionarily stable state (ESS) By definition, the necessary and sufficient condition for the state $\mathbf{x}^*$ to be

[5] In evolutionary games and its derivatives, it is at the aggregate level that a state (rather than strategy) can be mixed.

the ESS is for the inequality

$$\mathbf{x}^*\mathbf{A}\mathbf{x} > \mathbf{x}\mathbf{A}\mathbf{x} \tag{2}$$

to hold for all $\mathbf{x} \neq \mathbf{x}^*$ in the neighborhood of $\mathbf{x}^*$ in Γ^n. When an ESS, $\mathbf{x}^*$, is to be intruded by a state $\mathbf{x} \neq \mathbf{x}^*$ with meta-frequency ϵ, the inequality

$$\epsilon\mathbf{x}^*\mathbf{A}\mathbf{x} + (1-\epsilon)\mathbf{x}^*\mathbf{A}\mathbf{x}^* > \epsilon\mathbf{x}\mathbf{A}\mathbf{x} + (1-\epsilon)\mathbf{x}\mathbf{A}\mathbf{x}^*$$

must be satisfied for $\mathbf{x}^*$ to dominate the intruder $\mathbf{x}$. By rewriting this, we find

$$(1-\epsilon)\left[\mathbf{x}^*\mathbf{A}\mathbf{x}^* - \mathbf{x}\mathbf{A}\mathbf{x}^*\right] + \epsilon\left[\mathbf{x}^*\mathbf{A}\mathbf{x} - \mathbf{x}\mathbf{A}\mathbf{x}\right] > 0. \tag{3}$$

In the limit as ϵ approaches zero, (3) becomes equivalent to inequality (1), the definition of the Nash equilibrium. In a special case of the Nash equilibrium where $\mathbf{x}^*\mathbf{A}\mathbf{x}^* = \mathbf{x}\mathbf{A}\mathbf{x}^*$ for some $\mathbf{x} \neq \mathbf{x}^*$, (3) becomes equivalent to inequality (2), which is the definition of the ESS. Intuitively, if some frequency distribution other than the Nash equilibrium's frequency distribution is as optimal on the equilibrium, then the Nash equilibrium's frequency distribution must be superior to the other distribution on all the neighborhood points so that the state will be brought back to the equilibrium.

To allow this model to involve dynamics, evolutionary game theorists often employ the replicator equation (Taylor and Johnker, 1978; Schuster and Sigmund, 1983). The rate of growth (or of adoption) of a certain strategy is defined by the relative optimality of the performance of strategy i, namely the payoff for taking pure strategy i minus the mean payoff in the game world. Hence, it is expressed as

$$(\dot{\log x_i}) = \mathbf{e}_i\mathbf{A}\mathbf{x} - \mathbf{x}\mathbf{A}\mathbf{x}, \tag{4}$$

or equivalently,

$$\dot{x}_i = x_i(\mathbf{e}_i\mathbf{A}\mathbf{x} - \mathbf{x}\mathbf{A}\mathbf{x}). \tag{5}$$

This represents the standard form of the replicator equation when the payoff function is linear with matrix $\mathbf{A}$. From (4), we get

$$(\dot{\log x_i}) - (\dot{\log x_j}) = \mathbf{e}_i\mathbf{A}\mathbf{x} - \mathbf{e}_j\mathbf{A}\mathbf{x}.$$

Thus,

$$(\dot{\frac{x_i}{x_j}}) = (\frac{x_i}{x_j})(\mathbf{e}_i\mathbf{A}\mathbf{x} - \mathbf{e}_j\mathbf{A}\mathbf{x}), \tag{6}$$

provided $x_j \neq 0$. In the equilibrium, the ratio of a strategy's frequency to the other's frequency stays constant, or the time differential of the ratio is zero. So (6) indicates that $\mathbf{x}$ will be an equilibrium in int$\mathit{\Gamma}^n$ if and only if it satisfies

$$\mathbf{e}_i\mathbf{A}\mathbf{x} = \mathbf{e}_j\mathbf{A}\mathbf{x} \quad \forall i, j \quad \text{where } \mathbf{x} \in \mathit{\Gamma}^n. \tag{7}$$

The concepts presented so far concerns the dynamics in continuous time. Dekel and Scotchmer (1992) and Cabrales and Sobel (1992) present the versions for discrete time. The formula

$$x_i^{t+1} = x_i^t \frac{\mathbf{e}_i\mathbf{A}\mathbf{x}^t + \xi^t}{\mathbf{x}^t\mathbf{A}\mathbf{x}^t + \xi^t} \tag{8}$$

can be considered as the least objectionable candidate for the replicator equation in discrete time. A constant ξ^t is selected so that $\mathbf{e}_i\mathbf{A}\mathbf{x}^t + \xi^t$ will always be positive. Yet, our interpretation of ξ^t will be rather different, as will be discussed later (see Theorem 2). This discrete version of the replicator equation does not convey all the properties of the continuous version. Like the case where we compute ordinary differential equations using the Euler's method, periodic cycles which would be observed in (5) will be lost and the trajectories will converge to bdΓ^n when (8) is substituted for (5). However, (8) plays an important role when the theory discussed in this section is associated with ABMs involving Monte Carlo learning. The main reason is that ABMs intrinsically assume discrete time.

2.2 Agent-based models with Monte Carlo learning

Computational economics that uses ABMs has borrowed an idea from artificial intelligence, in that an agent perceives the state of the world, processes the information using internal functions, and returns a strategy - "action" is the term in AI - that optimizes the current and/or delayed payoff to himself (most general AI textbooks start with this concept, eg. Russell and Norvig, 1995). The algorithms for agents' internal functions vary. It could be simple condition-action rules, network-based regressions, statistical learning, or evolutionary algorithms. In our study, Monte Carlo learning is analyzed because of its strong relationship with the theory discussed in the previous section. When multiple agents are put in the game world, an agent's interaction with the world implicitly means his interaction with other agents. This logic justifies the use of ABMs for experimentations of games.

An agent could choose the optimal strategy, i, for the next time step such that

$$i = \arg\max_{j \in \mathtt{P}} \int_{\mathbf{x}^{t+1} \in \mathit{\Gamma}^n} \mathit{pdf}(\mathbf{x}^{t+1}) \cdot \mathbf{e}_j \mathbf{A} \mathbf{x}^{t+1},$$

where $\mathit{pdf}(\cdot)$ is probability density function. However, the problem here is that he may not have perfect information of $\mathit{pdf}(\mathbf{x}^{t+1})$ for all $\mathbf{x}^{t+1} \in \mathit{\Gamma}^n$, which he would roughly learn from life experiences. In our model, a simple AI is used for agents to learn directly the mapping of strategy-state pairs into expected payoff set[6] in the following way. Let $\mathit{\Omega} = \{(i, \mathbf{x}^t) \mid i \in \mathtt{P}, \mathbf{x}^t \in \mathit{\Gamma}^n\}$ be the set of $(n+1)$-tupple parameters of strategy-state pairs, and $\mathtt{V} = \{v^{t+1} \in \mathtt{R}^1\ \}$ be the set such that v^{t+1} is the prediction of the value of $\mathbf{e}_i \mathbf{A} \mathbf{x}^{t+1}$. Generally, an agent's internal function will be given in the form

$$F : \mathit{\Omega} \rightarrow \mathtt{V}. \tag{9}$$

This can be considered as prediction, since the agent expects that strategy i will cause the next period's payoff of $\mathbf{e}_i \mathbf{A} \mathbf{x}^{t+1}$ to turn out, having observed $\mathbf{x}^t$ in the current period. Alternatively, a recency-weighted observation of $(1-\lambda)\sum_{k=1}^{t} \lambda^{t-k-1} \mathbf{x}^k$ may be substituted for $\mathbf{x}^t$ of $\mathit{\Omega}$ in (9). While this sort of function is often realized by regression models or neural networks, let us adopt a discretized state model (tabular state space) for simplicity. Suppose that $\mathit{\Gamma}^n$ is separated into non-overlapping subsets such that $\bigcup_l \mathit{\Gamma}_l^n = \mathit{\Gamma}^n$ and $\mathit{\Gamma}_l^n \cap \mathit{\Gamma}_m^n = \emptyset$ for all $m \neq l$, which obviously implies $\mathit{\Gamma}_l^n \subset \mathit{\Gamma}^n \forall l$. Preferably,

[6] It is sound to believe that real human agents learn the direct mapping rather than probability. In Bayesian updating, learning for probability rather than direct mapping is concerned.

each Γ^n_l should have equal size. Define the new set $\bar{\Omega} = \{(i, l) \mid i \in \mathtt{P}, \Gamma^n_l \subset \Gamma^n\}$, and with a discrete indexing by l, (9) can be rewritten as

$$\bar{F} : \bar{\Omega} \to \mathtt{V}. \tag{10}$$

This function exhibits a minor weakness, in that all the states $\mathbf{x}$ belonging to category l are considered identical. However, it has an advantage when learning occurs from sampling. Given the function, agents will take strategy $i = \arg\max_k \bar{F}(k, l)$. Occasional explorations of not taking the optimal strategies are also important, for the reason that agents have to experience all the i-l combinations so that they enable $\bar{F}$ to be effective for most situations, if not all.

Learning, in this context, refers to the process of modifying function (10) so that it will return more and more accurate prediction values. To make writing simple, let $v_{i,l} = \bar{F}(i, l)$ denote the predicted value returned by the function $\bar{F}(i, l)$. That is, if the learning is fast enough (and if the equilibrium is not very unstable), we expect that $v_{i,l}$ eventually converges to $\mathbf{e}_i\mathbf{A}\mathbf{x}^{t+1}$ for $\mathbf{x}^t \in \Gamma^n_l$ in the limit as t approaches positive infinity. To introduce Monte Carlo learning, assume for the moment that an agent always perceives l and takes strategy i. With this assumption, the estimated value of $v_{i,l}$ at time t by average of samples is

$$v^t_{i,l} = \sum_{k=1}^{t} \mathbf{e}_i\mathbf{A}\mathbf{x}^k / t.$$

This is straightforward, yet the agent may run up his memory in this case, ie. he has to memorize all the payoffs in the past t time steps. An alternative learning rule equivalent to the above one is

$$v_{i,l}^{t} = \frac{1}{t}[\mathbf{e}_i \mathbf{A} \mathbf{x}^t - v_{i,l}^{t-1}] + v_{i,l}^{t-1}. \tag{11}$$

Now, the agent needs to memorize only two terms for learning. (11) executes equally weighted averaging of all the past payoffs. This algorithm works well if the world is deterministic in that the transition from $\mathbf{x}^t$ to $\mathbf{x}^{t+1}$ is guaranteed. However, the learning with (11) will be always obsolete if the world is stochastic. To make (11) a recency-weighted learning, we substitute a constant $\alpha \in (0,1)$ for $1/t$ in (11) so we have

$$v_{i,l}^{t} = \alpha[\mathbf{e}_i \mathbf{A} \mathbf{x}^t - v_{i,l}^{t-1}] + v_{i,l}^{t-1}. \tag{12}$$

The larger the value of α, the more recency-weighted is learning. This is evidenced by the following logic:

$$\begin{aligned} v_{i,l}^{t} &= \alpha[\mathbf{e}_i \mathbf{A} \mathbf{x}^t - v_{i,l}^{t-1}] + v_{i,l}^{t-1} \\ &= \alpha \mathbf{e}_i \mathbf{A} \mathbf{x}^t + (1-\alpha)\alpha \mathbf{e}_i \mathbf{A} \mathbf{x}^{t-1} + (1-\alpha)^2 \alpha \mathbf{e}_i \mathbf{A} \mathbf{x}^{t-2} + \cdots \\ &\approx \sum_{u=1}^{t} \alpha(1-\alpha)^{t-u} \mathbf{e}_i \mathbf{A} \mathbf{x}^u. \end{aligned}$$

This parameter for the degree of recency-weighting, α, may be associated with what Roth and Erev (1995) would refer to as the degree of “forgetting,” though their views might be slightly different. Besides, it also accounts for what Friedman (1998) refered to as “inertia” which is one of the most significant properties of evolutionary games. From another viewpoint, α works as the parameter to control the behaviors observed in macro dynamics, as will be discussed later. Conversely, we may calibrate α by the backward computation from the data of an observed macro dynamics.

It is unrealistic to assume that an agent always perceives l and takes strategy i. Unlike (11), recency-weighted learning (12) relaxes this assumption. Thus, it is safe to apply the general game representations to the recency-weighted Monte Carlo learning. In summary, an agent queries his function $\bar{F}$ for the expected payoff of taking strategy i in state l, and chooses such i that maximizes $v_{i,l}^t$, which is his estimate of $\mathbf{e}_i\mathbf{A}\mathbf{x}^{t+1}$. At the same time, he modifies $\bar{F}$ by using (12) so that it will return more accurate values in the future.

3 Connecting multi-agent Monte Carlo learning to the replicator equation

In the previous section, agents' learning was formalized at an individual level. We need a slight modification of the model in order to extend the algorithm to the analyses of aggregate behaviors. With the assumption that one-step transition among $\Gamma_1^n, \Gamma_2^n, \cdots$ makes small differences, the negative effects from eliminating our distinction of $v_{i,l}^t$ by l will be offset by the introduction of distribution-based notation of v_i^t.

Definition 1. *We define the estimated payoff of strategy $i \in \mathtt{P}$ weighted by its frequency as the total value estimate of strategy i. Let $\bar{v}_i^t$ denote the total value of v_i^t, which is the estimate of the total value of $\mathbf{e}_i\mathbf{A}\mathbf{x}$, namely $x_i\mathbf{e}_i\mathbf{A}\mathbf{x}$.*

(The following describes the rationale for employing the total value estimates for the aggregate model. It is only among that frequency of population, x_i^t, that strategy i is “believed” to be the maximizer of $\mathbf{e}_i\mathbf{A}\mathbf{x}^t$. Hence, the population of at least and at most this frequency (x_i^t) will encounter the payoff of $\mathbf{e}_i\mathbf{A}\mathbf{x}^t$ and update the function $\bar{F}$ of (10) with the error given by $\mathbf{e}_i\mathbf{A}\mathbf{x}^t - v_i^t$ for each agent. Hence, the aggregate error of this population can be defined as $x_i^t[\mathbf{e}_i\mathbf{A}\mathbf{x}^t - \frac{\bar{v}_i^t}{x_i^t}]\quad (= x_i^t\mathbf{e}_i\mathbf{A}\mathbf{x}^t - \bar{v}_i^t)$.) With the individual Monte Carlo learning (12) being modified for the total value estimate, the aggregate Monte Carlo learning can be defined as

$$\bar{v}_i^{t+1} := \alpha[x_i^t\mathbf{e}_i\mathbf{A}\mathbf{x}^t - \bar{v}_i^t] + \bar{v}_i^t. \tag{13}$$

For these definitions of aggregated Monte Carlo learning, we put the following sound assumptions:

Assumption 1: $i = \arg\max_j \mathbf{e}_j\mathbf{A}\mathbf{x}^t \Rightarrow x_i^t < \frac{1}{n}$

Assumption 2: $\min_i \bar{v}_i^t < \frac{\sum_j \bar{v}_j^t}{n}$

The first assumption states that the strategy that causes the least payoff will attract less than the average frequency. The second assumption states that the total value estimate of the smallest payoff is less than the average of total value estimates. Now, let variable $\delta(\bar{\mathbf{v}}^t)$ as a function of $\bar{\mathbf{v}}^t = (\bar{v}_1^t, \bar{v}_2^t, \cdots, \bar{v}_n^t)$ denote some value that is related to the distribution of $\bar{\mathbf{v}}^t$. With this definition, the following formula can be hypothesized as an estimate of state transition for the aggregate behavior of Monte Carlo agents.

$$x_i^{t+1} := \frac{\bar{v}_i^{t+1} + \delta(\bar{\mathbf{v}}^t)}{\sum_j(\bar{v}_j^{t+1} + \delta(\bar{\mathbf{v}}^t))}. \tag{14}$$

The endogenous variable $\delta(\bar{\mathbf{v}}^t)$ must satisfy the condition that $\bar{v}_i^{t+1} + \delta(\bar{\mathbf{v}}^t)$ be positive for all i.

Given the definition of the aggregate recency-weighted Monte Carlo learning (13), the hypothesized rule (14) will transform as follows.

$$\begin{aligned} x_i^{t+1} &:= \frac{\bar{v}_i^{t+1} + \delta(\bar{\mathbf{v}}^t)}{\sum_j (\bar{v}_j^{t+1} + \delta(\bar{\mathbf{v}}^t))} \\ &= \frac{\alpha x_i^t \cdot \mathbf{e}_i \mathbf{A} \mathbf{x}^t + (1-\alpha)\bar{v}_i^t + \delta(\mathbf{v}^t)}{\sum_j (\alpha x_j^t \cdot \mathbf{e}_j \mathbf{A} \mathbf{x}^t + (1-\alpha)\bar{v}_j^t + \delta(\mathbf{v}^t))} \\ &= x_i^t \cdot \frac{\mathbf{e}_i \mathbf{A} \mathbf{x}^t + \frac{(1-\alpha)\bar{v}_i^t + \delta(\bar{\mathbf{v}}^t)}{\alpha x_i^t}}{\mathbf{x}^t \mathbf{A} \mathbf{x}^t + \frac{\sum_j (1-\alpha)\bar{v}_j^t + n\delta(\bar{\mathbf{v}}^t)}{\alpha}}. \end{aligned} \tag{15}$$

Let us define ϕ^t and ψ^t as

$$\begin{aligned} \phi^t &= \frac{(1-\alpha)\bar{v}_i^t + \delta(\bar{\mathbf{v}}^t)}{\alpha x_i^t} \quad \text{and} \\ \psi^t &= \frac{\sum_j (1-\alpha)\bar{v}_j^t + n\delta(\bar{\mathbf{v}}^t)}{\alpha}, \end{aligned} \tag{16}$$

thus enabling (15) to be written in the simple form

$$x_i^{t+1} = x_i^t \cdot \frac{\mathbf{e}_i \mathbf{A} \mathbf{x}^t + \phi^t}{\mathbf{x}^t \mathbf{A} \mathbf{x}^t + \psi^t}. \tag{17}$$

The update rule (17) thus resembles the discrete replicator equation (8).

Lemma 1 *(8) and (17) (and thus (14)) are equivalent if and only if* $\phi^t = \psi^t = \xi^t$

Proof Substitute ξ^t for ϕ^t and ψ^t in (17), and the sufficiency is obvious. Necessity by contrapositive: if $\xi^t \neq \phi^t \vee \xi^t \neq \psi^t$, then (8) and (17) cannot be equivalent. □

Since (8) is (merely) suggested as a candidate (Hofbauer and Sigmund, 1998), (17) would still represent the discrete replicator equation even if $\phi^t \neq \psi^t$ as long as appropriate normalization is executed on $\mathbf{x}^{t+1}$. Yet, our study sticks to the case where ϕ^t equals ψ^t. By equating ϕ^t to ψ^t in (16), we get the endogenous variable

$$\delta(\bar{\mathbf{v}}^t) = (\alpha - 1)\frac{\bar{v}_i^t - x_i^t \sum_j \bar{v}_j^t}{1 - x_i^t n}. \tag{18}$$

For (17) to be equivalent to (8), this equation of the endogenous variable (18) must be equal for a given t for all i.

Lemma 2 *The endogenous variable (18) for a given t is equal for all $i \in \mathtt{P}$ in the game.*

In order to prove this lemma, consider two cases: (a) $\mathbf{x}$ is an equilibrium at the center of $\mathit{\Gamma}^n$; and (b) other cases. (case (b) covers (a) as well.)

Proof - **Case (a)**: $\mathbf{x}$ is an equilibrium at the center of $\mathit{\Gamma}^n$

If the frequency distribution is an equilibrium, $\mathbf{e}_i\mathbf{A}\mathbf{x} = \mathbf{e}_j\mathbf{A}\mathbf{x}$ $(= c)$ holds from (7), where we use c to denote the corresponding value. With the definition of the aggregate recency-weighted Monte Carlo learning (13), this translates to $\bar{v}_i^t - (1-\alpha)\bar{v}_i^{t-1} = \alpha x_i c$ for all i. Using this equation, (18) can be rewritten as

$$\begin{aligned}
\delta(\bar{\mathbf{v}}^t) &= (\alpha - 1)(1-\alpha)\frac{\bar{v}_i^{t-1} - x_i^t \sum_j \bar{v}_j^{t-1}}{1 - x_i^t n} + x_i\alpha(\alpha-1)c \\
&= (\alpha - 1)(1-\alpha)^2\frac{\bar{v}_i^{t-2} - x_i^t \sum_j \bar{v}_j^{t-2}}{1 - x_i^t n} \\
&\qquad + x_i\alpha(\alpha-1)(1-\alpha)c + x_i\alpha(\alpha-1)c \\
&\quad \vdots \\
&= (\alpha - 1)(1-\alpha)^t\frac{\bar{v}_i^{0} - x_i^t \sum_j \bar{v}_j^{0}}{1 - x_i^t n} - x_i\alpha c\sum_{k=1}^{t}(1-\alpha)^k \\
&= -(1-\alpha)^{t+1}\frac{\bar{v}_i^{0} - x_i^t \sum_j \bar{v}_j^{0}}{1 - x_i^t n} - x_i c[(1-\alpha) - (1-\alpha)^{t+1}].
\end{aligned}$$

Since $\lim_{t\to\infty}(1-\alpha)^{t+1}=0$, this eventually can be simplified to

$$\delta(\bar{\mathbf{v}}^t)=-x_i c(1-\alpha). \tag{19}$$

For the case where $\mathbf{x}$ is at the center of $\mathit{\Gamma}^n$, we have $x_i = x_j (=\frac{1}{n})$ for all i and j. The endogenous variable $\delta(\bar{\mathbf{v}}^t)$ turns out to be independent of strategy if $\mathbf{x}$ is an equilibrium at the center of $\mathrm{int}\mathit{\Gamma}^n$. □

From (14) in conjugate with (18) and (19), any underestimated total values of payoff on average ($\bar{v}_i^t \quad \forall i$) cannot fall short of $\mathbf{e}_i\mathbf{A}\mathbf{x}(1-\alpha)/n$.

Proof - **Case (b)**: other cases

First, define β_{ij} such that $\bar{v}_j^t=\bar{v}_i^t\cdot\beta_{ij}$ for all i and j. Obviously, β_{ij} equals β_{ji}^{-1}. Then, (18) can be written as

$$\delta(\bar{\mathbf{v}}^t)=(\alpha-1)\bar{v}_i^t\cdot\frac{1-x_i^t\sum_j\beta_{ij}}{1-x_i^t n}.$$

Variable $\delta(\bar{\mathbf{v}}^t)$ is equal for all the n strategies if and only if

$$\bar{v}_i^t\cdot\frac{1-x_i^t\sum_j\beta_{ij}}{1-x_i^t n}=\bar{v}_k^t\cdot\frac{1-x_k^t\sum_j\beta_{kj}}{1-x_k^t n}, \tag{20}$$

for all i and k. By the way, the next two equations are true from our definitions.

$$\begin{aligned}&\frac{\bar{v}_i^t}{\bar{v}_k^t}=\frac{\sum_j\beta_{kj}}{\sum_j\beta_{ij}} \quad \text{and}\\ &\sum_i x_i^t=1. \end{aligned} \tag{21}$$

One way to ensure our argument is to show that (21) is sufficient for (20). We will show this only for the case of $n=2$. Since we have (21) or $1-x_1^t-x_2^t=0$,

$$\bar{v}_1^t(1-x_1^t-x_2^t)=\bar{v}_2^t(1-x_1^t-x_2^t)=0 \quad \text{or}$$

$$\begin{aligned}
\bar{v}_1^t(1+x_1^t-2x_1^t-x_2^t) &= \bar{v}_2^t(1+x_2^t-x_1^t-2x_2^t) \quad \text{or} \\
\frac{\bar{v}_1^t+x_1^t(\bar{v}_1^t+\bar{v}_2^t)-2\bar{v}_1^t x_1^t}{\bar{v}_1^t+\bar{v}_2^t} &= \frac{\bar{v}_2^t+x_2^t(\bar{v}_1^t+\bar{v}_2^t)-2\bar{v}_2^t x_2^t}{\bar{v}_1^t+\bar{v}_2^t} \quad \text{or} \\
\frac{\bar{v}_1^t+x_1^t(\bar{v}_1^t+\bar{v}_2^t)-n\bar{v}_1^t x_1^t}{\bar{v}_1^t+\bar{v}_2^t} &= \frac{\bar{v}_2^t+x_2^t(\bar{v}_1^t+\bar{v}_2^t)-n\bar{v}_2^t x_2^t}{\bar{v}_1^t+\bar{v}_2^t}. \qquad (22)
\end{aligned}$$

Now, in order to eliminate $\bar{v}_1^t$ and $\bar{v}_2^t$ from (22), we use the following equations derived from our definition.

$$\begin{aligned}
\frac{\bar{v}_1^t+\bar{v}_2^t}{\bar{v}_1^t} &= \frac{\bar{v}_1^t}{\bar{v}_1^t}+\frac{\bar{v}_2^t}{\bar{v}_1^t} = \sum_j \beta_{1j}, \quad \text{and similarly,} \\
\frac{\bar{v}_1^t+\bar{v}_2^t}{\bar{v}_2^t} &= \sum_j \beta_{2j}.
\end{aligned}$$

Given this, (22) can be rewritten as

$$\begin{aligned}
\frac{1+x_2^t(\sum_j \beta_{1j}-n)}{\sum_j \beta_{1j}} &= \frac{1+x_1^t(\sum_j \beta_{2j}-n)}{\sum_j \beta_{2j}} \quad \text{or} \\
(\sum_j \beta_{2j})(1-x_2^t-x_1^t\sum_j \beta_{1j}) &= (\sum_j \beta_{1j})(1-x_1^t-x_2^t\sum_j \beta_{2j}) \quad \text{or} \\
(\sum_j \beta_{2j})(1-x_2^t-x_1^t\sum_j \beta_{1j} &+ x_1^t x_2^t n\sum_j \beta_{1j}) \\
&= (\sum_j \beta_{1j})(1-x_1^t-x_2^t\sum_j \beta_{2j}+x_1^t x_2^t n\sum_j \beta_{2j}) \quad \text{or} \\
\frac{\sum_j \beta_{2j}}{\sum_j \beta_{1j}} &= \frac{1-x_2^t\sum_j \beta_{2j}}{1-x_1^t\sum_j \beta_{1j}} \cdot \frac{1-x_1^t n}{1-x_2^t n}.
\end{aligned}$$

By relating (21) and the above equation, we get

$$\begin{aligned}
\frac{\bar{v}_1^t}{\bar{v}_2^t} &= \frac{1-x_2^t\sum_j \beta_{2j}}{1-x_1^t\sum_j \beta_{1j}} \cdot \frac{1-x_1^t n}{1-x_2^t n} \quad \text{or} \\
\bar{v}_1^t \cdot \frac{1-x_1^t\sum_j \beta_{1j}}{1-x_1^t n} &= \bar{v}_2^t \cdot \frac{1-x_2^t\sum_j \beta_{2j}}{1-x_2^t n},
\end{aligned}$$

which exhibits exactly the same form as (20). □

Since ξ^t has the domain $(-\min_i \mathbf{e}_i\mathbf{A}\mathbf{x}^t, \infty)$, we can have $\phi^t = \psi^t = \xi^t$ if and only if the proposition of the following lemma is true.

Lemma 3 *We have* $\mathbf{e}_i\mathbf{Ax}^t + \phi^t > 0$, $\mathbf{e}_i\mathbf{Ax}^t + \psi^t > 0 \forall i$ *for those conditions specified in Table 1.*

Proof Since the implication of Lemma 2 is $\phi^t = \psi^t \; \forall i$, we can use (18) for this proof, and we only need to show $\mathbf{e}_i\mathbf{Ax}^t + \phi^t > 0 \; \forall i$. It is clear that $\mathbf{e}_i\mathbf{Ax}^t + \phi^t > 0 \; \forall i \iff \min_i \mathbf{e}_i\mathbf{Ax}^t + \phi^t > 0$. Let subscript i denote $\arg\min_j \mathbf{e}_j\mathbf{Ax}^t + \phi^t$. By substituting (18), the value of ϕ^t in (16) will be

$$\phi^t = \frac{1}{\alpha(1 - x_i^t n)}[(1-\alpha)\sum_j \bar{v}_i^t - (1-\alpha)n\bar{v}_i^t].$$

The addition of $\mathbf{e}_i\mathbf{Ax}^t$ to the above equation yields

$$\begin{aligned}\mathbf{e}_i\mathbf{Ax}^t + \phi^t &= \frac{1}{\alpha(1 - x_i^t n)}[\alpha(1 - x_i^t n)\mathbf{e}_i\mathbf{Ax}^t + (1-\alpha)(\sum_j \bar{v}_j^t - n\bar{v}_i^t)] \\ &= \frac{\alpha\tilde{e}_i^t + (1-\alpha)\tilde{\phi}_i^t}{\alpha(1 - x_i^t n)},\end{aligned}$$

where $\tilde{e}_i^t = (1 - x_i^t n)\mathbf{e}_i\mathbf{Ax}^t$ and $\tilde{\phi}_i^t = \sum_j \bar{v}_j^t - n\bar{v}_i^t$. By Assumption 1, the denominator is positive, implying the equivalence between the positivity of $\mathbf{e}_i\mathbf{Ax}^t + \phi^t$ and that of $\alpha\tilde{e}_i^t + (1-\alpha)\tilde{\phi}_i^t$ (numerator). Besides, this tells that $\tilde{e}_i^t \gtreqless 0 \Leftrightarrow \mathbf{e}_i^t\mathbf{Ax}^t \gtreqless 0$. Similarly, by Assumption 2, $\tilde{\phi}^t > 0$ holds.

(A) *when* $\mathbf{e}_i\mathbf{Ax}^t > 0$: Since $\alpha\tilde{e}_i^t + (1-\alpha)\tilde{\phi}_i^t$ is a convex set in $\mathtt{R}^1$ with $\tilde{e}_i^t$ and $\tilde{\phi}_i^t$ as end points, we have $\mathbf{e}_i\mathbf{Ax}^t > 0 \Rightarrow \tilde{e}_i^t > 0 \Rightarrow \mathbf{e}_i\mathbf{Ax}^t + \phi^t > 0$. This satisfies the first column of Table 1.

(B) *when* $\mathbf{e}_i\mathbf{Ax}^t \leq 0$ *and* α *is relatively low:* The inequality $\alpha\tilde{e}_i^t + (1-\alpha)\tilde{\phi}_i^t > 0$ is equivalent to $\alpha < \frac{\tilde{\phi}_i^t}{\tilde{\phi}_i^t - \tilde{e}_i^t}$. Thus, all $\alpha < \frac{\tilde{\phi}_i^t}{\tilde{\phi}_i^t - \tilde{e}_i^t}$ (that are relatively low, but greater than 0) will satisfy $\mathbf{e}_i\mathbf{Ax}^t + \phi^t > 0$ for the condition corresponding to the first row - second column of Table 1.

		$\min_i \mathbf{e}_i A x^t$	
		> 0	≤ 0
α	relatively low	Yes	Yes
	relatively high	Yes	Yes/No

Table 1 The conditions in which Lemma 3 and Theorem 1 do and do not hold.

(C) *when* $\mathbf{e}_i \mathbf{A}\mathbf{x}^t \leq 0$ *and* α *is relatively high*: From (B), all $\alpha \leq \frac{\tilde{\phi}_i^t}{\tilde{\phi}_i^t - \tilde{e}_i^t}$ fail to satisfy $\mathbf{e}_i \mathbf{A}\mathbf{x}^t + \phi^t > 0$. If $\frac{\tilde{\phi}_i^t}{\tilde{\phi}_i^t - \tilde{e}_i^t} \leq 1$, we have "No" for the second row - second column of Table 1. However, if $\frac{\tilde{\phi}_i^t}{\tilde{\phi}_i^t - \tilde{e}_i^t}$ is greater than one or is outside of the domain of α, then we have "Yes" for all the entries of Table 1. □

Finally, we arrive at the following theorem, which is the main claim of this paper.

Theorem 1 *The discrete replicator equation (8) and the aggregate model of distributed Monte Carlo learning (14) are equivalent for those conditions specified in Table 1.*

Proof Since ξ^t in (8) is the parameter to be freely selected in $(-\min_i \mathbf{e}_i \mathbf{A}\mathbf{x}^t, \infty)$, it holds that $\phi^t = \psi^t > -\min_i \mathbf{e}_i \mathbf{A}\mathbf{x}^t \implies \exists \xi^t : \phi^t = \psi^t = \xi^t$. Lemma 2 implies $\phi^t = \psi^t \; \forall i$. Given this and Lemma 3, there exists some value of ξ^t such that $\phi^t = \psi^t = \xi^t > -\min_i \mathbf{e}_i \mathbf{A}\mathbf{x}^t$ for those conditions specified in Table 1. Hence, from Lemma 1, we conclude the truth of the proposition of Theorem 1. □

Equation (14) with the endogenous variable defined in (18) represents the aggregate Monte Carlo rule that is equivalent to the discrete replicator equation (8) for those conditions specified in Table 1. Equation (14) may be considered as the medium of the agent based models and the replicator equation that, we expect, will give rise to a formal process in which computational and theoretical models will be fused. Surely, it relies on the assumptions made at the beginning of this section.

4 Experiments: how aggregate fluctuations are related to the degree of individuals' recency-weighting

Having seen the theoretical aspect of the relationship between the two models, one might be tempted to observe and compare the simulation results of them. Let us take a simple zero-sum game as an example. The payoff matrix is defined as

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}. \tag{23}$$

In this case, it is concluded from (4) that

$$\begin{aligned} (\dot{\log} x_1) + (\dot{\log} x_2) + (\dot{\log} x_3) &= 0 \quad \text{or} \\ \frac{d}{dt}(\log x_1 x_2 x_3) &= 0 \quad \text{or} \\ x_1 x_2 x_3 &= \text{constant}. \end{aligned} \tag{24}$$

Hence, the trajectories draw periodic cycles around the equilibrium point in $\mathrm{int}\varGamma^3$ with $x_1x_2x_3$ being a constant of motion. Fig. 1 (a) depicts typical motions of periodic cycle in $\mathrm{int}\varGamma^3$. If we assume continuous time, equation (5) can be used to simulate the dynamics of (23). An experiment with the

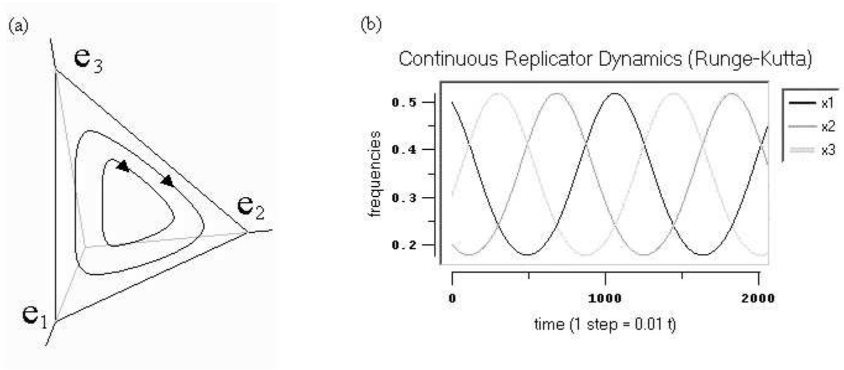

Fig. 1 (a) Trajectories of periodic cycles where the game is a special case of zero-sum game. (b) A result of simulation using the continuous replicator equation with fourth-order Runge Kutta method.

fourth-order Runge-Kutta method gives a result illustrated by Fig. 1 (b). Clearly, it follows a periodic cycle that agrees with (24) and Fig. 1 (a).

However, as briefly mentioned earlier, the assumption of discrete time makes the consequences quite different. Let us assume that the parameter ξ is sufficiently large such that $\mathbf{e}_i\mathbf{A}\mathbf{x}$ be positive for all i. Note that $\mathbf{x}\mathbf{A}\mathbf{x}$ is always zero for (23). The Nash equilibrium (1/3, 1/3, 1/3) is not the ESS since $\mathbf{x}^*\mathbf{A}\mathbf{x} - \mathbf{x}\mathbf{A}\mathbf{x} = 0$ (see (2)). Now, let us define the function

$$V(\mathbf{x}) = \prod_{i=1}^{3} x_i^{x_i^*},$$

and we have

$$\frac{\dot{V}}{V} = \sum_{i=1}^{3} x_i^* \frac{\dot{x}_i}{x_i}. \tag{25}$$

While equation (25) assumes continuous time, we can approximate the value of $\dot{x}_i$ for the discrete version (8) by using the leapfrog method, as

$$\dot{x}_i \approx \frac{1}{2}(x_i^{t+1} - x_i^{t-1})$$

$$
\begin{aligned}
&= \frac{1}{2}[x_i^t \frac{\mathbf{e}_i\mathbf{A}\mathbf{x}^t + \xi^t}{\mathbf{x}^t\mathbf{A}\mathbf{x}^t + \xi^t} - x_i^t \frac{\mathbf{x}^{t-1}\mathbf{A}\mathbf{x}^{t-1} + \xi^{t-1}}{\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1} + \xi^{t-1}}] \\
&= \frac{x_i^t}{2}[\frac{\mathbf{e}_i\mathbf{A}\mathbf{x}^t}{\xi^t} + \frac{\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1}}{\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1} + \xi^{t-1}}].
\end{aligned}
$$

Substitute this in (25), and we get

$$
\begin{aligned}
\frac{\dot{V}}{V} &\approx \frac{1}{2}\sum_{i=1}^{3} x_i^*[\frac{\mathbf{e}_i\mathbf{A}\mathbf{x}^t}{\xi^t} + \frac{\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1}}{\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1} + \xi^{t-1}}] \\
&= \frac{1}{6}\sum_{i=1}^{3} \frac{\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1}}{\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1} + \xi^{t-1}}.
\end{aligned}
$$

Since

$$
\sum_{i=1}^{3} \frac{\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1}}{\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1} + \xi^{t-1}} < 0 \quad \forall \xi^{t-1} > -\min_i \mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1}, \quad \mathbf{x} \neq \mathbf{x}^*,
$$

it turns out that

$$
\frac{\dot{V}}{V} < 0, \qquad \dot{V} < 0.
$$

Function V is a negative gradient-like Lyapunov function. Thus, all non-equilibrium states in int$\mathit{\Gamma}^3$ will converge to bd$\mathit{\Gamma}^3$. However, this may not be the case when ξ^t takes a large value, since we have

$$
\lim_{\xi \to \infty} \dot{V} = 0. \tag{26}
$$

For a reasonably large value of ξ^t, the discrete replicator dynamics will behave like the continuous version. Additionally, we find

$$
\begin{aligned}
\frac{d}{d\xi^{t-1}}(\frac{\dot{V}}{V}) &= -\frac{1}{6}\sum_{i=1}^{3} \frac{\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1}}{(\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1} + \xi^{t-1})^2} \\
&> 0 \quad \forall \xi^{t-1} > -\min_i \mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1}, \quad \mathbf{x} \neq \mathbf{x}^*.
\end{aligned}
$$

Since $\dot{V}/V < 0$, this means that the larger values of ξ^t will lead to flatter gradients.

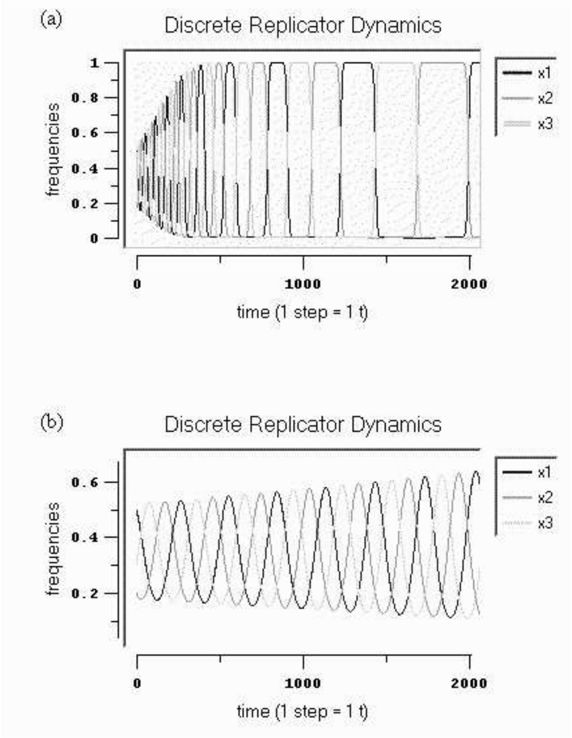

Fig. 2 Results of simulation using the discrete replicator equation (a) for $\xi^t = 5.0$, and (b) for $\xi^t = 25.0$.

Fig. 2 (a) is a result of (23) with the discrete replicator equation (8) for $\xi^t = 5.0$, and Fig. 2 (b) for $\xi^t = 25.0$. States gradually converge to the boundary subset of Γ^3, and the period of cycles becomes longer as time passes. Smaller value of ξ^t causes more rapid convergence. More formally, we arrive at the following theorem.

Theorem 2 *The parameter ξ^t of the discrete replicator equation (8) is likely to be inversely related with the degree of agents' recency-weighting, α.*

Proof We ideally need the assumption that the total value estimates are not affected by α, as if they were exogenous (Appendix B presents the difficulty without this assumption; however, this assumption is valid for many cases, as presented in Appendix B). Then, from (16), we have

$$\begin{aligned}
\frac{\partial \phi^t}{\partial \alpha} &= -x_i^t[\bar{v}_i^t + \delta(\bar{\mathbf{v}}^t)]/(\alpha x_i^t)^2 \\
&\approx -x_i^t[\bar{v}_i^{t+1} + \delta(\bar{\mathbf{v}}^t)]/(\alpha x_i^t)^2 < 0, \qquad \text{and similarly} \\
\frac{\partial \psi^t}{\partial \alpha} &= -\sum_j[\bar{v}_j^t + \delta(\bar{\mathbf{v}}^t)]/\alpha^2 \\
&\approx -\sum_j[\bar{v}_j^{t+1} + \delta(\bar{\mathbf{v}}^t)]/\alpha^2 < 0.
\end{aligned}$$

Since Theorem 1 and Lemma 1 hold, we are likely to have $\partial\xi^t/\partial\alpha < 0$ too.

□

Intuitively, smaller ξ^t causes agents to respond more sensitively to the current state (obvious from (8)), and agents with greater degree of recency-weighting, α, behave in a similar way. Theorem 2 in conjugate with $\dot{V} < 0$ and $\frac{d}{d\xi^t}(\frac{\dot{V}}{V}) > 0$ for the special case of (23) implies that a group of more recency-weighting agents tends to generate rapid growth of wave amplitude, which is graphically evidenced by Fig. 2 and Fig. 3. This makes sense also from the following example. Assume that the states $\{\mathbf{x} \mid x_1 > x_2 \approx x_3\}$ continue for some duration (eg. $t = 100$ to 110). At $t = 111$, the ordering of $v_3 > v_1 > v_2$ will prevail among most (if not all) agents if they put heavy weights on the recent experiences due to (12). On the other hand, less recency-weighting agents do not necessarily make such temporarily biased ordering depending on what experiences they had from $t = 0$ to 99. In this

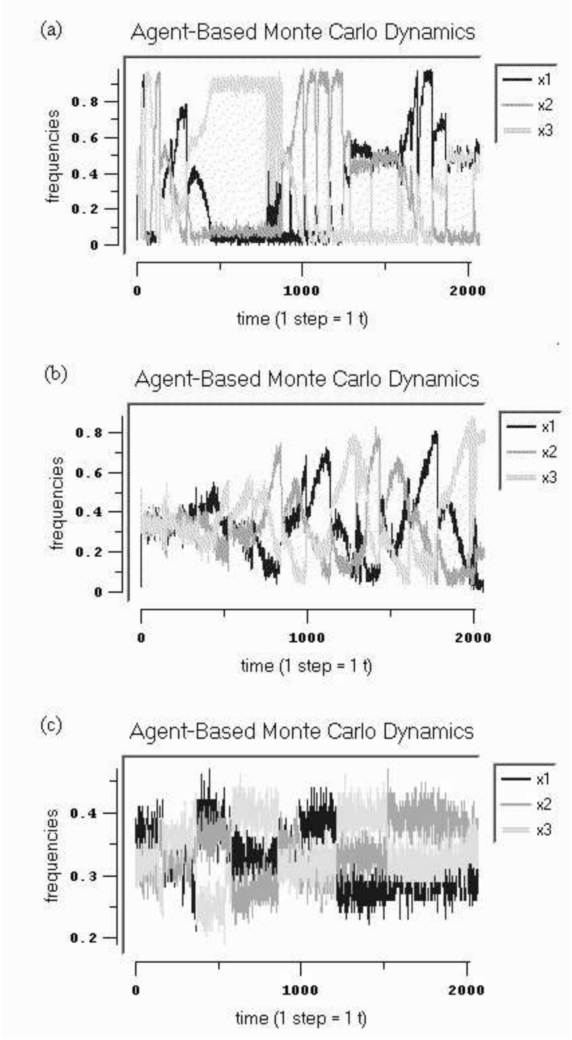

Fig. 3 Results of simulation using multi-agent Monte Carlo learning (a) for more recency-weighting parameter ($\alpha = 0.5$), (b) for less recency-weighting parameter ($\alpha = 0.1$), and (c) for the least recency-weighting parameter ($\alpha = 0.00001$).

case, a group of more recency-weighting agents tends to lean toward $\mathbf{e}_3$ more than a group of less recency-weighting agents. In the next phase, this will be the case with $\mathbf{e}_2$, and so on. Hence, as stated in Sec. 2.2, α can be used as a controlable parameter that affects macro dynamics, and conversely, we can calibrate α from the data of an observed meso or macro scale dynamic.

Similar phenomena can be observed in multi-agent simulation. Fig. 3 (a) shows a typical result of the simulation with multi-agent Monte Carlo learning for $\alpha = 0.5$, and Fig. 3 (b) for $\alpha = 0.1$. The trajectories are not such neat lines as numerical solutions because decision-making is made by each of autonomous agents, and the frequency vector only reflects the consequences of bottom-up processes. However, its pattern of convergence to $\mathrm{bd}\Gamma^t$ and periods of cycles resemble those of the discrete replicator dynamics. Notice that Fig. 2 (a) and (b) are analogous to Fig. 3 (a) and (b), respectively. This, again, evidences the relationship between parameter ξ^t and the agents' degree of recency-weighting α, the proposition of which is found in Theorem 2. Additionally, (26) and Theorem 2 imply that an extremely small value of α generates periodic cycles, almost like the continuous replicator dynamics. This is also evidenced by a result of the simulation with multi-agent Monte Carlo learning for $\alpha = 0.00001$, shown in Fig. 3 (c). Notice that Fig. 3 (c) is analogous to Fig. 1 in its appearance[7], rather than Fig. 2. Hence, the

[7] For the convenience to restrict our concern to the deterministic evolutionary game, it is stated so. However, if the topic were extended to include stochastic evolutionary games, this result would need to be associated with the stochastic dynamics presented by Foster and Young (1990).

distributed Monte Carlo learning model can be related not only with the discrete replicator equation (8), but also with the continuous version (5).

5 Conclusions and discussions

A tight relationship between the multi-agent simulation with Monte Carlo learning and the prototype evolutionary game theory with replicator equations was proved and experimentally evidenced. This not only shows the similarity of the dynamics in these two models. It also formalizes the process by which micro behaviors of autonomous agents translate to the aggregate dynamics of a society at large (ie. fluctuation pattern in macro dynamics was derived from micro factors of recency-weighting.). This result provides a basis for prospective theories bridging micro and macro dynamics models. Moreover, it demonstrates that the experimental results of simulations with distributed intelligence can be backed up by theories to a reasonable extent. An additional contribution of the proofs of Theorem 1 and Theorem 2 is to illustrate the intuitively mysterious constant ξ^t in (8). The relationship between the ABMs and the replicator model has added an interpretation of ξ^t as being associated with the degree of agents' recency-weighting. This suggests the possibility that the seemingly chaotic behaviors of ABMs may be explained by theoretical models with varying degrees of recency weighting.

The consequence of this paper poses an interesting question. The ABMs' side of the evolution is considered active, since learning based on expected payoff optimization is the result of intentional computation. On the other hand, the nature of prototype evolutionary game is considered to be passive at the micro level, due to the premise made by most biologists and ecologists that natural selection drives evolution that relies mostly on chance. If the results presented in this paper are valid (ie. the definition and assumptions made in Sec. 3 are appropriate), then this implies that active behaviors of economic agents and passive being (such as genes) give rise to the equivalent dynamics. At this point, there is no sufficient logic with this to affect the recent disputes on the validity of biological metaphors in evolutionary economics. As far as economics is concerned, we could argue that the dominance of evolutionary process might vanish the macro effects of agent-wise activeness. In this case, the question arises "how big of a role does micro activeness play in determining the macro dynamics?" This clearly depends on the complexity of agent interactions, or the network structure. We have analyzed the effects of a micro factor, α, on macro dynamics. However, this is far from all.

As mentioned previously, Monte Carlo learning is one of the most fundamental algorithms to decide the process of agent learning. For example, Sutton and Barto (1998) extended Monte Carlo learning to develop major reinforcement learning algorithms, such as Q-learning and Sarsa. With computational reinforcement learning, an agent can learn the future-cumulative payoff of a strategy by bootstrapping the expected values of those states that the strategy he chose will stochastically lead to. With this property of reinforcement learning, agents will cope more effectively with repeated games than Monte Carlo agents can. Interestingly, a slight modification to equation (12) with a flavor of the Bellman equation or dynamic programming (Bellman, 1957) will realize some reinforcement learning algorithms. Thus, the aggregate reinforcement learning will be easily defined with some forms similar to (15). Our study leaves open extensions to these and many other learning algorithms. It is expected that a series of such works on complex systems will help bridge the computational and theoretical fields, and as well as micro and macro dynamics models.

Appendix A: The Algorithm of Agent-Based Monte Carlo Learning

A simplified pseudocode for the agent-based Monte Carlo learning is presented below. The characters and symbols used here are consistent with those in the text.

A.1. Model program

```
-   repeat until program terminates
-       for each strategy $i$ do
-           $x_i^{t-1} \leftarrow x_i$
-           $x_i \leftarrow 0$
-       end do
-       for each $agent$ do
-           $i \leftarrow$ agent:chooseStrategy( $\{x_1^{t-1}, x_2^{t-1}, \cdots, x_n^{t-1}\}$ )
-           $x_i \leftarrow x_i + 1.0$ / totalNumberOfAgents
-           end do
-           for each strategy $i$ do
-                   $e_i Ax \leftarrow \sum_j a_{ij} \times x_j$
-           end do
-           for each *agent* do
-                   *agent:learn*( $\{e_1 Ax, e_2 Ax, \cdots, e_n Ax\}$ )
-               end do
-       end repeat
```

A.2. Strategy choice by an agent

```
-       agent:chooseStrategy(state)
-           $l \leftarrow$ categorize(state)
-           bestPayoff $\leftarrow -\infty$
-           for each strategy $i$ do
-                   if $v_{il} >$ bestPayoff then
-                           bestPayoff $\leftarrow v_{il}$
-                           strategy $\leftarrow i$
-                   end if
-           end do
-           return strategy
-       end
```

A.3. Learning by an agent

```
-       agent:learn(payoffSet)
-           for each strategy $i$ do
-           if $i = strategy$ then
-               $v_{il} \leftarrow \alpha[(e_i Ax \leftarrow payoffSet) - v_{il}] + v_{il}$
-           end if
-       end do
-   end
```

Appendix B: Path-Dependence Consideration on the Relationship between $\boldsymbol{\xi^t}$ and $\boldsymbol{\alpha}$

Since Lemma 1 and Theorem 1 hold, we treat ξ^t, ϕ^t, and ψ^t as identical parameters. Let us take ϕ^t as representative, and consider it as a function of α. Before analyzing the sensitivity of ϕ^t to α, we need to take a look at that of $\bar{v}_i^t$. From (13), we derive

$$
\begin{aligned}
\frac{\partial \bar{v}_i^t}{\partial \alpha} &= [x_i^{t-1}\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1} - \bar{v}_i^{t-1}] + (1-\alpha)\frac{\partial \bar{v}_i^{t-1}}{\partial \alpha} \\
&= [x_i^{t-1}\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-1} - \bar{v}_i^{t-1}] \\
&\quad +(1-\alpha)[x_i^{t-2}\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-2} - \bar{v}_i^{t-2}] + (1-\alpha)^2\frac{\partial \bar{v}_i^{t-2}}{\partial \alpha} \\
&\quad \vdots \\
&\approx \sum_{k=0}^{t}(1-\alpha)^k[x_i^{t-k-1}\mathbf{e}_i\mathbf{A}\mathbf{x}^{t-k-1} - \bar{v}_i^{t-k-1}].
\end{aligned}
$$

Intuitively, this value represents the recency-weighted cumulative errors of aggregate estimates, or aggregate reminiscence of past errors. Here, the errors are in terms of the degree of underestimates. Let $ARPE_i^t$ (standing for Aggregate Reminiscence of Past Errors) denote the value of $\partial\bar{v}_i^t/\partial\alpha$.

$ARPE$ is truely chaotic and path-dependent unless $\alpha = 1^-$ or the dynamics is at least locally stable.

Now let us turn to the sensitivity of ϕ^t to α. From (16), we derive

$$\begin{aligned}\frac{\partial \phi^t}{\partial \alpha} &= -\frac{x_i^t[\bar{v}_i^t + \delta(\bar{\mathbf{v}}^t)]}{(\alpha x_i^t)^2} + \frac{1-\alpha}{\alpha x_i^t}[\frac{\partial \bar{v}_i^t}{\partial \alpha} - \sum_j \frac{1-x_j^t}{1-nx_j^t}\frac{\partial \bar{v}_j^t}{\partial \alpha}] \\ &= -\frac{x_i^t[\bar{v}_i^t + \delta(\bar{\mathbf{v}}^t)]}{(\alpha x_i^t)^2} + \frac{1-\alpha}{\alpha x_i^t}[ARPE_i^t - \sum_j \frac{1-x_j^t}{1-nx_j^t} ARPE_j^t].\end{aligned}$$

We know $\lim_{\alpha \to 1} d\phi^t/d\alpha = 0$. Thus, it is necessary that $d^2\phi^t/d\alpha^2 < 0$ for the domain of α to have the inverse relationship with ξ^t. However, $d^2\phi^t/d\alpha^2$ depends more sophisticatedly on the path-dependent terms of $ARPE$'s. Hence, we simply make a weak argument as presented in Theorem 2. Fortunately, we may find many cases where $ARPE_i^t$ eventually vanishes, such as the dynamics with exponential stability, oscillatory stability, and exponential unstability due to the boundedness of Γ^n (major counter-examples are oscillatory unstability and limit cycles). In such cases, Theorem 2 will be stronger.

Acknowledgement

The excellent supervisions and advice given by Dr. Arthur J. Caplan, Utah State University, have greatly contributed to this paper.

References

1. Arthur BW (1993) On designing economic agents that behave like human agents. Journal of Evolutionary Economics 3: 1-22

2. Barto AG, Singh SP (1990) On the computational economics of reinforcement learning. Proceedings of the 1990 Connectionist Models Summer School, San Mateo, 35-44

3. Bellman RE (1957) Dynamic programming. Princeton University Press, Princeton

4. Cabrales A, Sobel J (1992) On the limit points of discrete selection dynamics. Journal of Economic Theory 57: 407-419

5. Cressman R (1992) The stability concept of evolutionary game theory. Springer Verlag, Berlin

6. Dekel E, Scotchmer S (1992) On the evolution of optimizing behavior. Journal of Economic Theory 57: 392-346

7. Dewdney AK (1984) Computer recreations sharks and fish wage an ecological war on the toroidal planet wa-tor. Scientific American, December 1984, 14-22

8. Dosi G, Marengo L, Fagiolo G (1996) Learning in evolutionary environments. University of Trento, Computable and Experimental Economics Laboratory Paper # 9605

9. Erev I, Roth AE (1995) Predicting how people play games: reinforcement learning in experimental games with unique, mixed strategy equilibria. American Economic Review 88: 848-881

10. Foster D, Young PH (1990) Stochastic Evolutionary Game Dynamics. Theoretical Population Biology 38: 219-232

11. Friedman D (1991) Evolutionary games in economics. Econometrica 59: 637-666

12. Friedman D (1998) On economic applications of evolutionary game theory. Journal of Evolutionary Economics 8: 15-43

13. Fudenberg D, Levine DK (1998) The theory of learning in games. MIT Press, Cambridge

14. Haefner JW (1996) Modeling biological systems: principles and applications. Chapman & Hall, New York

15. Hofbauer J (1981) On the occurrence of limit cycles in the Volterra-Lotka equation. Nonlinear Analysis 5: 1003-1007

16. Hofbauer J, Sigmund K (1988) The theory of evolution and dynamical systems. Cambridge University Press, Cambridge

17. Hofbauer J, Sigmund K (1998) Evolutionary games and population dynamics. Cambridge University Press, Cambridge

18. Hofbauer J, Sigmund K (2003) Evolutionary game dynamics. Bulletin of the American Mathematical Society, 40: 479-519

19. Holland JH, Miller JH (1991) Artificial adaptive agents in economic theory. American Economic Review, Papers and Proceedings 81: 365-370

20. Judd KL (1997) Computational economics and economic theory: complements or substitutes? Journal of Economic Dynamics and Control 21: 907-942

21. Judd KL (2001) Computation and economic theory: introduction. Economic Theory 18: 1-6

22. Kandori M, Mailath GJ, Rob R (1993) Learning, mutation, and long run equilibria in games. Econometrica 61: 29-56

23. Langton CG (1989) Artificial life. In: Langton CG (ed), Artificial life. Santa Fe Institute Studies in the Science of Complexity, 1-44, Addison-Wesley, Redwood City

24. Maes P (1995) Modeling adaptive autonomous agents. In: Langton CG (ed) Artificial life: an overview, 135-162. MIT Press, Cambridge

25. Maynard Smith J (1974) The theory of games and the evolution of animal conflicts. Journal of Theoretical Biology 47: 209-221

26. Roth AE, Erev I (1995) Learning in extensive-form games: experimental data and simple dynamic models in the intermediate term. Games and Economic Behavior. Special Issue: Nobel Symposium 8: 164-212

27. Russell SJ, Norvig P (1995) Artificial intelligence: a modern approach. Prentice-Hall, Upper Saddle River

28. Samuelson L, Zhang J (1992) Evolutionary stability in asymmetric games. Journal of Economic Theory 57: 363-391

29. Schuster P, Sigmund K (1983) Replicator dynamics. Journal of Theoretical Biology 100: 533-538

30. Sutton RS, Barto AG (1998) Reinforcement learning. MIT Press, Cambridge

31. Swinkels J (1993) Adjustment dynamics and rational play in games. Games and Economic Behavior 5: 455-484

32. Tesfatsion L (2000) Agent-based computational economics: a brief guide to the literature, Discussion Paper, Economics Department, Iowa State University, Jan. 2000, Reader's Guide to the Social Sciences, Fitzroy-Dearborn, London

33. Tesfatsion L (2002) Agent-based computational economics: growing economics from the bottom up. Artificial Life 8: 55-82

34. Taylor PD, Jonker L (1978) Evolutionarily stable strategies and game dynamics. Mathematical Bioscience 40: 145-156

35. von Neumann J (1966) The theory of self-reproducing automata. University of Illinois Press, Urbana

36. Weibull JW (1997) Evolutionary game theory. MIT Press, Cambridge